World Models: Foundations, Applications, and Limitations

Deep Learning Techniques for Sequential Decision Making

World Models: Foundations, Applications, and Limitations

Deep Learning Techniques for Sequential Decision Making

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Friday, 06 February 2026 at 10:00 o'clock

by

Cristian MEO

Master of Science in Mechanical Engineering, Delft University of Technology,
Netherlands born in Catania, Italy

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Dr. ir. J.H.G. Dauwels,	Delft University of Technology, *promotor*
Prof. dr. ir. G.J.T. Leus,	Delft University of Technology, *promotor*
Independent members:	
Prof. dr. ir. M. Wisse,	Delft University of Technology
Dr. C. Della Santina,	Delft University of Technology
Dr. F. Fioranelli,	Delft University of Technology
Dr. P.L. Lanillos,	Consejo Superior de Investigaciones Científicas
Prof. dr. ir. A.J. van der Veen,	Delft University of Technology, *reserve member*

Keywords: Generative Modeling, Video Prediction, World Modeling, Representation Learning

ISBN 973-24-1218-342-5

An electronic copy of this dissertation is available at
https://repository.tudelft.nl/.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

The ability to predict and model the future is a cornerstone of intelligence, underpinning decision-making and adaptation in dynamic environments. Building intelligent machines carries the potential to advance automation and increase living standards around the world. Recent advancements in deep learning have enabled algorithms to make accurate predictions when large datasets of examples are available, facilitating classification and generation of images and text. Despite this remarkable progress, algorithms still struggle when few examples are available, such as for controlling robots or encountering unforeseen situations. Unlike most learning algorithms, humans quickly adapt to unseen scenarios and learn new skills from relatively small amounts of experience. This ability stems, in part, from internal models of the world, which allow humans to imagine future outcomes of potential actions. Teaching machines to learn world models accurate enough for successful planning has been challenging, especially when dealing with large unstructured inputs such as videos.

This dissertation addresses the challenge of building artificial intelligence systems that can anticipate future states, adapt to novel scenarios, and make robust decisions with limited data by investigating autoregressive deep state-space models for video prediction and world modeling. At its core, the dissertation focuses on two interconnected tasks: the challenge of accurately forecasting future frames in video prediction settings, where pixel-level fidelity and understanding motion, object interactions, and physical rules are crucial, and the broader problem of modeling an environment's dynamics via actions-latents causal relationships. Even small inaccuracies in predicted frames can compound over extended sequences, creating significant deviations from reality.

Through the lens of model-based reinforcement learning, this dissertation demonstrates that internal rollouts generated by a learned world model can guide action selection, dramatically reducing the number of actual environment interactions needed to reach competent or even expert-level performance. This property is particularly valuable for robotics and other high-stakes domains, where data acquisition can be slow, expensive, or dangerous. To improve the learning behaviour of video prediction and world models, this dissertation presents several inductive biases, such as objective functions that encourage time consistency between frames or that help modeling extreme events, a masked generative prior that improves the sequence modelling capabilities of the dynamics modules, disentangled representations that improve exploration strategies, physics-informed approaches to incorporate physical constraints, and attention-based workspaces to enhance multi-agent coordination.

Although the proposed methods present performance gains in various experimental setups, the real value of this dissertation lies in the versatility of the proposed inductive biases. These biases, built and evaluated across different domains, are

designed with the potential for application in large-scale architectures, suggesting that the same algorithmic principles can be repurposed for vision-driven control tasks or for anticipating rare climate events with potentially large societal impacts. These findings bridge a variety of application domains, from simple simulated environments to real-world tasks, illustrating how breakthroughs in generative modeling and self-supervised learning can be systematically harnessed to tackle the complexity of dynamic scenes and interactive decision-making. Improved data efficiency also reduces the environmental footprint of large-scale training regimes.

In conclusion, this dissertation answers research questions that highlight the transformative potential of generative and latent modeling frameworks to reshape how machines perceive, learn about, and ultimately act within the environments they encounter. By bridging latent imagination, generative representations, and self-supervised objectives, this work reveals a path toward artificial systems that not only learn rapidly from experience but also exhibit interpretability and generalization capabilities, bringing us closer to intelligent agents capable of robust, forward-looking reasoning and collaboration.

SAMENVATTING

Het vermogen om de toekomst te voorspellen en te modelleren is een hoeksteen van intelligentie, die ten grondslag ligt aan besluitvorming en aanpassing in dynamische omgevingen. Het bouwen van intelligente machines heeft het potentieel om de automatisering te bevorderen en de levensstandaard wereldwijd te verhogen. Recente vorderingen in deep learning hebben algoritmen in staat gesteld om nauwkeurige voorspellingen te doen wanneer grote datasets met voorbeelden beschikbaar zijn, waardoor classificatie en generatie van afbeeldingen en tekst mogelijk zijn. Ondanks deze opmerkelijke vooruitgang hebben algoritmen nog steeds moeite wanneer er weinig voorbeelden beschikbaar zijn, zoals voor het besturen van robots of het tegenkomen van onvoorziene situaties. In tegenstelling tot de meeste leeralgoritmen passen mensen zich snel aan onbekende scenario's aan en leren ze nieuwe vaardigheden van relatief weinig ervaring. Dit vermogen komt deels voort uit interne modellen van de wereld, die mensen in staat stellen om zich toekomstige uitkomsten van potentiële acties voor te stellen. Het plannen in de verbeelding stelt mensen in staat beslissingen te nemen zonder alle mogelijke strategieën in de echte wereld uit te proberen. Het is echter een uitdaging gebleken om machines wereldmodellen te leren die nauwkeurig genoeg zijn voor succesvolle planning, vooral bij het omgaan met grote, ongestructureerde inputs zoals video's.

Dit proefschrift behandelt de uitdaging van het bouwen van kunstmatige intelligentiesystemen die toekomstige toestanden kunnen anticiperen, zich kunnen aanpassen aan nieuwe scenario's en robuuste beslissingen kunnen nemen met beperkte data door het onderzoeken van autoregressieve deep state-space modellen voor videovoorspelling en wereldmodellering. De kern van het proefschrift richt zich op twee onderling verbonden taken: de uitdaging van het nauwkeurig voorspellen van toekomstige frames in videovoorspellingsinstellingen, waarbij pixel-niveaugetrouwheid en begrip van beweging, objectinteracties en fysische regels cruciaal zijn, en het bredere probleem van het modelleren van de dynamiek van een omgeving via acties-latente causale relaties. Zelfs kleine onnauwkeurigheden in voorspelde frames kunnen zich over langere sequenties opstapelen, wat leidt tot aanzienlijke afwijkingen van de werkelijkheid.

Door de lens van modelgebaseerd reinforcement learning demonstreert dit proefschrift dat interne uitrol gegenereerd door een geleerd wereldmodel de actieselectie kan sturen, waardoor het aantal daadwerkelijke omgevingsinteracties dat nodig is om competent of zelfs expertniveau te bereiken, drastisch wordt verminderd. Deze eigenschap is met name waardevol voor robotica en andere risicovolle domeinen, waar data-acquisitie traag, duur of gevaarlijk kan zijn. Om het leergedrag van videovoorspellings- en wereldmodellen te verbeteren, presenteert dit proefschrift verschillende inductieve biases, zoals objectieve functies die tijdsconsistentie tussen

frames aanmoedigen of die helpen bij het modelleren van extreme gebeurtenissen, een gemaskeerd generatief prior dat de sequentiemodelleringsmogelijkheden van de dynamische modules verbetert, ontwarde representaties die exploratiestrategieën verbeteren, fysisch-geïnformeerde benaderingen om fysische beperkingen op te nemen, en op aandacht gebaseerde werkruimten om de coördinatie in multi-agentsystemen te stroomlijnen en te verbeteren.

Hoewel de voorgestelde methoden prestatiewinst laten zien in verschillende experimentele opstellingen, ligt de werkelijke waarde van dit proefschrift in de veelzijdigheid van de voorgestelde inductieve biases. Deze biases, gebouwd en geëvalueerd in verschillende domeinen, zijn ontworpen met het potentieel voor toepassing in grootschalige architecturen, wat suggereert dat dezelfde algoritmische principes kunnen worden hergebruikt voor visueel gestuurde controletaken of voor het anticiperen op zeldzame klimaatevenementen met potentieel grote maatschappelijke gevolgen. Deze bevindingen overbruggen een verscheidenheid aan toepassingsdomeinen, van eenvoudige gesimuleerde omgevingen tot real-world taken, en illustreren hoe doorbraken in generatieve modellering en zelf-supervisie systematisch kunnen worden ingezet om de complexiteit van dynamische scènes en interactieve besluitvorming aan te pakken. Verbeterde data-efficiëntie vermindert ook de ecologische voetafdruk van grootschalige trainingsregimes.

Concluderend beantwoordt dit proefschrift onderzoeksvragen die het transformatieve potentieel van generatieve en latente modelleringskaders benadrukken om de manier waarop machines hun omgeving waarnemen, erover leren en er uiteindelijk in handelen, te hervormen. Door latente verbeelding, generatieve representaties en zelf-supervisieobjectieven te overbruggen, onthult dit werk een pad naar kunstmatige systemen die niet alleen snel leren van ervaring, maar ook interpreteerbaarheid en generalisatievermogen vertonen, waardoor we dichter bij intelligente agenten komen die in staat zijn tot robuuste, toekomstgerichte redenering en samenwerking.

PREFACE

It was the end of my first year when I realized that embodied intelligence would be the key to define a meaningful PhD trajectory. It all started with my internship at Mila, a period of excitement and struggles. A researcher from Google DeepMind, Anirudh Goyal, after referencing me for a position supervised by Prof. Yoshua Bengio, Turing Winner 2019 and founder of MILA, the Quebec AI Institute, introduced me to model-based reinforcement learning, or world modelling. The idea of predicting how the world unfolds, and acting over these predictions to achieve meaningful goals, reminded me of my whole life. Planning ahead the next three moves, trying to get the next thing done as efficiently as possible. It reminded me of the five-year plan I made when I started my bachelor's degree, the way I live every single day, the way intelligence works - I dare say.

Although the learning curve was very steep, among several failures I managed to make some meaningful contributions to the research community, publishing in some of the most influential conferences and establishing meaningful collaborations. For this, I need to thank my supervisor, Prof. Justin Dauwels, who supported every step I took along the way.

While this journey has been very successful, it has also been full of challenges, insecurities and struggles. I would have never achieved such accomplishments without the constant support of my family, especially of my mother Pinella, who has been my lighthouse whenever everything else was dark. I would have never made it through sixty-hours work schedules without my friends being always there, cheering for my every success, listening to my elaborate plans willingly and, sometimes, unwillingly, bringing joy and laughter every single day. Particularly, I would like to thank my flatmates, who eventually became a second family, and our whole group, with whom I shared most of my happiest moments in recent years.

I would also like to thank some people, who trusted and dedicated their time to support my ambition and my goals. Antal Baggerman, who has been and still is one of the greatest mentors I ever had. Anirudh Goyal, who proved to me that life doesn't have to be a zero-sum game. Alfio Tosto, who taught me about philosophy and inspired me to start my academic journey. Alejandro, who trusts me enough to start a business with me, I hope to live up to your expectations and ambitions!

Finally, I would like to thank my girlfriend, Isabela, who managed to make me fall in love. Thank you for supporting and taking care of me all the time.

Cristian Meo
Delft, May 2025

INTRODUCTION

The future influences the present as much as the past.

Friedrich Nietzsche

The ability to predict and model the future is a cornerstone of intelligence, underpinning decision-making and adaptation in dynamic environments [1]. Building intelligent machines carries the potential to advance automation and increase living standards around the world [2]. The pursuit of artificial intelligence (AI) has motivated generations of scientists and driven technological breakthroughs, including the invention of the computer [3]. Recent advancements in deep learning have enabled algorithms to make accurate predictions when large datasets of examples are available, facilitating the classification and generation of images and text [4, 5]. Despite this remarkable progress, algorithms still struggle when few examples are available, such as for controlling robots or encountering unforeseen situations [6, 7]. Unlike most learning algorithms, humans quickly adapt to unseen scenarios and learn new skills from relatively small amounts of experience. This ability stems, in part, from internal models of the world, which allow humans to imagine the future outcomes of potential actions [1]. Planning in imagination enables humans to make decisions without trying out all possible strategies in the real world. However, teaching machines to learn world models able to perform successful planning has been challenging, especially when dealing with large, unstructured inputs such as videos [8].

1. PROBLEM DEFINITION

1.1. VIDEO PREDICTION

Video prediction is the task of forecasting future frames of a video sequence given a series of past frames. Formally, let

$$\mathbf{x}_{1:T} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$$

denote a sequence of T observed video frames, where each frame $\mathbf{x}_t \in \mathbb{R}^{H \times W \times C}$ for height H, width W, and channels C. The goal is to predict

$$\mathbf{x}_{T+1:T+K} = (\mathbf{x}_{T+1}, \dots, \mathbf{x}_{T+K}),$$

a sequence of K future frames as shown in Fig. 1. One common approach is to learn a parametric model $p_\theta(\mathbf{x}_{T+1:T+K} \mid \mathbf{x}_{1:T})$ that captures the conditional distribution of future frames given the past, where θ represents the trainable parameters of the

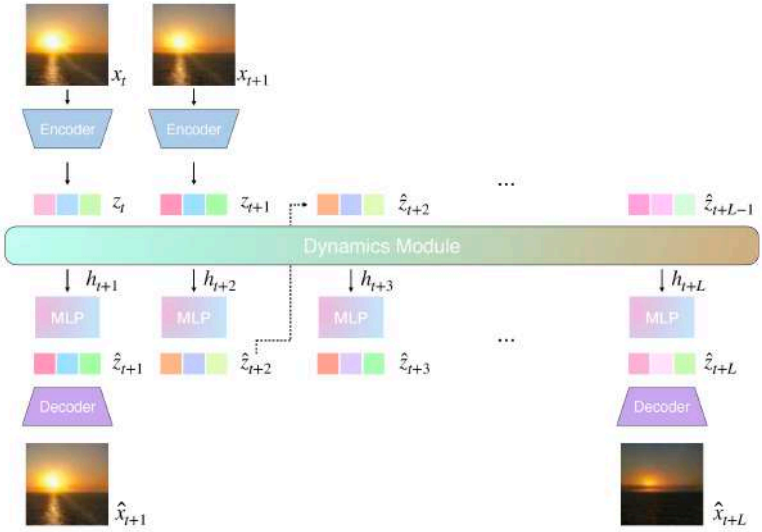


Figure 1: The image shows the general video prediction pipeline used throughout the thesis. Images $x_{1:T}$ are encoded into latent representations $z_{1:T}$ that are used to predict future states using a latent dynamics module. The predicted latents are then decoded to output future frames.

model [9]. Learning this model is inherently challenging because of the extremely large input and output spaces, as each frame may contain thousands to millions of pixels, and because real-world videos encode intricate spatial structures and rich temporal dynamics. Predictions must remain coherent over extended sequences, and even small errors can accumulate, causing the predicted frames to deviate significantly from reality. Furthermore, even simple scenarios require an understanding of object interactions, motion dynamics, and contextual cues, all while maintaining consistency over multiple time steps.

1.2. WORLD MODELING

World modeling extends video prediction by constructing structured or abstract representations of the environment's dynamics [8]. Instead of merely predicting future frames at a pixel level, world modeling seeks to learn a representation $\mathbf{z}_t$ for each time step t, which captures the state of the environment in a more compact or semantically meaningful way. Formally, one can define a generative world model as follows:

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_t \mid \mathbf{z}_t)\, p_\theta(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}),$$

where $\mathbf{x}_t$ are observed frames (or other sensor inputs), and $\mathbf{z}_t$ is a latent state encoding the environment's status at time t [8], as shown in Fig. 2. The parameters

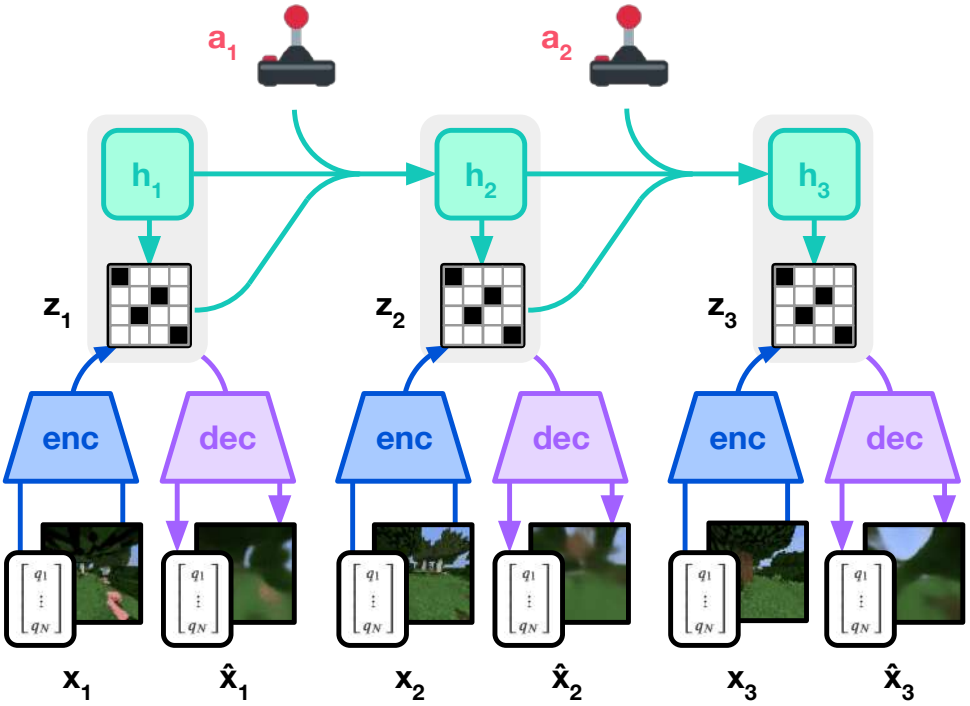


Figure 2: The world model encodes sensory inputs into discrete representations z_t that are predicted by a sequence model with recurrent state h_t given actions a_t. The inputs are reconstructed to shape the representations.

θ govern both the observation model (how the latent state generates frames) and the transition model (how the latent state evolves over time). Training world models is challenging because discovering meaningful, low-dimensional representations that capture the underlying physics and semantics of the environment is non-trivial, and the entire state of the world is often only partially observable in any single frame. Moreover, world models must be accurate enough to support decision-making and planning, because errors in the model can lead to suboptimal or unsafe actions when deployed. Finally, real-world environments exhibit vast variability, so a robust world model must generalise across tasks and adapt to previously unseen scenarios. These two problems—video prediction and world modeling—are closely related. Video prediction emphasizes pixel-level forecasting, while world modeling seeks higher-level representations that can be used for planning and reasoning about the future. Both require powerful generative models capable of capturing complex, high-dimensional data.

2. Challenges and State of the Art

2.1. Classical Forecasting and Deep Learning Paradigm

Until the deep learning era, non-trivial video prediction was widely viewed as intractable, as classical statistical methods struggled with both the combinatorial explosion of future states and the representational complexity inherent in high-dimensional data [10]. By contrast, the advent of deep neural networks, including convolutional architectures [11] and sequence models such as Recurrent Neural Networks (RNNs) [12] or Transformers [5], revolutionized the paradigm by leveraging large-scale datasets and end-to-end feature extraction pipelines. Still, retaining accuracy across long time horizons remains a formidable challenge, as compounding errors accumulate in sequence predictions, often demanding models that account for physical laws, object interactions, and causal relationships. Ambiguities in real-world video data mean there may be multiple plausible futures, making uncertainty quantification an ongoing research issue.

2.2. World Modeling and Latent Planning

The pursuit of *world modeling* seeks to give machines the ability to "imagine" future scenarios by learning environment dynamics from experience [8, 13]. Though the potential applications span from autonomous driving, climate modeling, robotics to other high-stake domains, data acquisition can be costly or infeasible, reinforcing the need for approaches that learn effectively from limited examples. Additionally, world models that perform well in one domain often need extensive retraining to transfer to another, highlighting the persistent gap in generalization.

2.3. Representation Learning and Generative Modeling

Within the broader landscape of human-like intelligence research, generative modeling paved the way for powerful data representation: Variational Autoencoders (VAEs) [14] and Generative Adversarial Networks (GANs) [15] pioneered strategies for capturing complex distributions, followed by sequence extensions such as Variational RNNs (VRNNs) [16]. More recently, Denoising Diffusion Probabilistic Models (DDPMs) and their score-based and latent variants have delivered state-of-the-art sample quality by iteratively reversing a noise-injection process in pixel or latent space [17–19]. Disentanglement learning [20] helped illuminate latent factors of variation (e.g., shape or pose), and self-supervised frameworks [2] began leveraging unlabelled data at scale. These advances resulted in a new wave of "world models", starting with Ha and Schmidhuber's demonstration that policy learning can benefit from latent imagination [13], subsequently inspiring agents such as Dreamer [21] and DayDreamer [22] that reduce physical interactions through internal trajectory simulation. Agents built on these ideas rely on effectively capturing environment dynamics from sensors to support accurate long-horizon predictions [23]. Early work like PlaNet [24] and the original World Models approach [13] illustrated how compact latent planning can significantly lower sample requirements.

One influential blueprint has been the Recurrent State Space Model (RSSM) [24], which splits latent representations into deterministic and stochastic parts. This

framework underlies the Dreamer series [21, 25, 26], delivering notable performance on continuous control benchmarks while using fewer environment interactions. Subsequent work tackled reconstruction-free learning to mitigate the overhead of pixel-level decoding [27–30]. Another thread emerged with LeCun's Joint-Embedding Predictive Architecture (JEPA) [31], which shifts emphasis from raw frame generation to higher-level embedding spaces; this principle drives a suite of self-supervised models like I-JEPA [32], A-JEPA [33], MC-JEPA [34], and V-JEPA [35].

2.4. WORLD MODELING

Transformers [5] have likewise catalyzed advances in generative modeling. Works such as TransDreamer [36], IRIS [37], TWM [38], and STORM [39] harness attention-based mechanisms to capture extended temporal dependencies. Google's Genie [40] extends this concept by learning generative interactive environments from massive unlabelled video, representing a move toward universal, manipulative world modeling. These architectures have proven fruitful in both gaming and robotics. For instance, Atari [41] remains a canonical testbed for reinforcement learning [42, 43], but model-free methods can demand millions of interactions, prompting approaches like SimPLe [44] that leverage stochastic video prediction to reduce sample overhead. DreamerV2 [25] refined discrete latent modeling for improved performance, and a series of Transformer-based world models [36–39] have since demonstrated even greater efficiency and, in some cases, human-level play with comparatively fewer training steps.

2.5. REAL-WORLD APPLICATIONS

In real-world robotics, the cost of environment interaction is amplified. As shown in Fig. 3, early methods such as PILCO [45] and dynamics-modeling approaches [46] assumed privileged access to ground-truth states, but PlaNet [24] and the Dreamer family [21, 25, 26] circumvented this by learning directly from pixels and planning in latent spaces for high-dimensional tasks within the DeepMind Control Suite [47]. Several advances removed the need for pixel reconstruction [27–29], strengthened safety constraints in uncertain settings [48], and balanced multiple objectives [49]. RoboDreamer [50] introduced compositional language-based world modeling for zero-shot adaptation, while UniSim [51] embeds agent actions into a generative simulator capable of real-world control. Works like DayDreamer [22] and Robot-DreamPolicy [52] have shown that with effective latent rollouts, real robotic systems can acquire and refine skills more rapidly, alleviating concerns about the physical costs of trial and error.

Beyond games and robots, other domains leverage world-model concepts for tasks like navigation. PathDreamer [53] integrates latent imagination to predict panoramic scenes in unexplored indoor regions, thereby improving navigation success. DreamerV3 [26] demonstrates cross-domain generality by maintaining fixed hyperparameters across Atari, DeepMind Control, and Minecraft, echoing the broader push toward universal agents. Plan2Explore [54] encourages self-supervised exploration, enabling a more comprehensive environment model prior to domain-specific fine-

tuning, and large-scale human-video datasets help discover richer affordances with minimal real-robot experience [55]. Despite rapid progress, the field still struggles with compounding errors in long-term prediction [29, 39], domain transfer hurdles, and the intricacies of designing interpretable, safe systems for deployment [48].

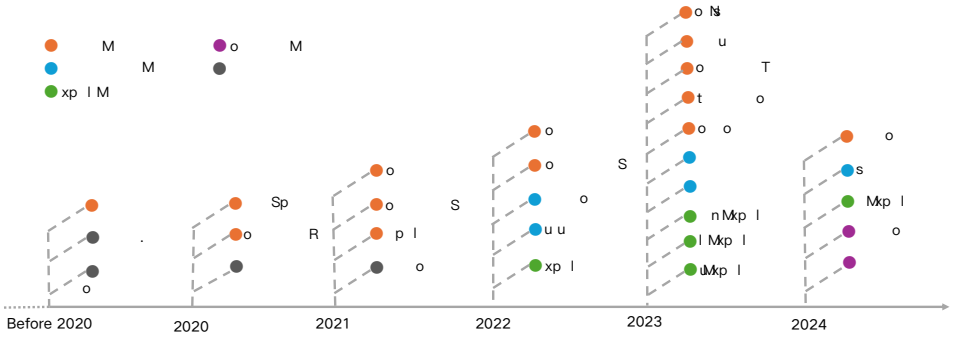


Figure 3: World Models Timeline Overview.

Overall, the increasing sophistication of generative and latent modeling frameworks, coupled with a surge of interest in Transformers and representation-driven approaches like JEPA, has brought predictive intelligence closer to practical real-world impact. Researchers are converging on strategies that balance realism with scalability, gleaned from self-supervised or weakly supervised signals, to mitigate data requirements and enhance robustness. As flexible architectures mature, the enduring aims include seamlessly handling multi-domain scenarios, integrating uncertainty and safety considerations, and ultimately delivering agents that can anticipate, reason, and act effectively in open-ended environments.

2.6. LIMITATIONS AND THESIS CONTRIBUTIONS

Despite impressive advances, today's world models share four key shortcomings:

Limited Generalization Across Tasks. Many models work well in a single game or environment but fail when moved to new tasks. Chapter 1 develops a representation learning method (α-TCVAE) that finds semantic, disentangled latent features. Such representations make models more data-efficient, while also improving their generalization capabilities.

Error Accumulation Over Time. When models predict many steps into the future, small mistakes compound and forecasts quickly become unrealistic. In Chapter 2, we propose a temporally consistent object-centric framework that handles this problem by keeping track of different objects across frames, hence reducing error propagation to localized areas, resulting in an overall improvement of long sequence generations.

Poor Handling of Rare, Extreme Events. Conventional methods rarely see extreme examples during training and struggle to recognise or predict them.

Chapter 3 introduces an extreme value loss function that specifically focuses on learning extreme precipitation events, allowing the model to learn their patterns from data instead of assuming fixed prototypes. Chapter 4 builds on this by enforcing physical laws, resulting in more realistic rainfall forecasts.

Poor Sequence Modeling Capabilities. A major limitation of existing transformer-based world models is that their autoregressive dynamics heads rely on simple multilayer perceptrons to predict the next latent state. This makes them both sample-inefficient, struggling to capture long-range dependencies, and difficult to be adopted for continuous control tasks. Chapter 5 addresses this by replacing the standard dynamics head with a Masked Generative Prior (MaskGIT). This change yields state-of-the-art performance on the Atari100k benchmark and, for the first time, extends transformer-based world models successfully to continuous action domains (DeepMind Control Suite), where it significantly improves both prediction accuracy and policy performance.

Poor Coordination of Embodied Agents. In cooperative multi-agent settings, optimal team behaviour depends on dynamically sharing relevant information and adapting when conditions change. Most existing Multi-Agent Reinforcement Learning (MARL) methods either ignore inter-agent communication or rely on fixed policy architectures that struggle under environmental variability. Chapter 6 presents the Stateful Active Facilitator (SAF), which equips agents with a shared "knowledge source" that filters and distributes only task-relevant signals during training, and a shared pool of specialized policies that agents can select from at runtime. This simple attention-based mechanism significantly improves coordination and adaptability, enabling agents to consistently achieve higher returns even as the environment's complexity and heterogeneity increase.

Together, these contributions directly target some of the most pressing gaps in current world models. Not only, improving long-term accuracy, extreme event forecasting, task generalisation, and computational efficiency, but also bringing us closer to AI systems that can reliably imagine, plan, and act in complex real-world settings.

3. THESIS OVERVIEW AND CONTRIBUTIONS

This thesis positions itself within this evolving landscape, contributing to the development of video prediction and world modeling as key enablers of Artificial General Intelligence (AGI). By focusing on disentanglement, temporal consistency, and multimodal integration, it aims to address fundamental challenges and push the boundaries of what these models can achieve. Through a series of interconnected studies, the thesis explores novel methods for improving the capabilities and applications of video prediction and world modeling, laying a foundation for intelligent systems that can learn, adapt, and reason about the world [8].

3.1. OBJECTIVES AND CONTRIBUTIONS

This thesis contributes to the advancement of video prediction, world modeling, and their role in achieving AGI through the following interconnected studies. For the sake of clarity, Fig. 4 shows how each chapter contributes to a specific part of the general autoregressive deep state space model.

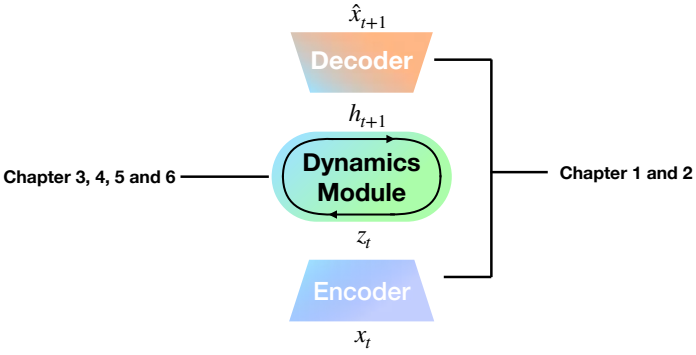


Figure 4: This diagram shows the architecture backbone used throughout the thesis and the contributions related to each module.

Chapter 1: Disentanglement and Diversity. The first chapter introduces α-TCVAE, investigating the relationship between disentanglement and diversity in generative models. This study introduces a new lower bound of the Total Correlation between input and learned latent representations that generalises the well-established β-VAE [20] and provides a more explainable interpretation of the objective function used to optimise generative feature extractors. Moreover, this paper shows how improving the generative capabilities of artificial agents can improve their ability to imagine a more diverse set of futures, improving their exploration capabilities and, ultimately, their downstream task performances.

Chapter 2: Temporal Consistency and Object-Centric Biases. The second chapter explores object-centric temporal consistency, leveraging conditional autoregressive inductive biases to enhance video prediction. This work improves the ability of object-centric models to capture temporal dynamics and maintain consistency across frames [9]. As a result, the proposed approach is able to improve video-related downstream tasks (e.g., VideoQA).

Chapter 3: Extreme Precipitation Nowcasting with Transformer

Models. The third chapter applies transformer-based generative models to nowcasting extreme precipitation events. This study demonstrates how imposing a meaningful inductive bias on the learned latent space can significantly improve the learning behaviour of a given model. Specifically, in this work, we propose an Extreme Value Loss that improves the model's capability of representing and predicting extreme precipitation events in the Netherlands.

Chapter 4: Physics-Informed Generative Models for Precipitation. The fourth chapter introduces physics-informed generative models for precipitation nowcasting. By integrating domain-specific knowledge, this study exemplifies how video prediction models can improve predictive accuracy [15], when informing the network of the physical laws describing dynamic state transitions of the considered data.

Chapter 5: Masked Generative Priors for Sequence Modeling. The fifth chapter examines how masked generative priors enhance the sequence modeling capabilities of world models. However, the contributions are relevant to any transformer-based autoregressive model (e.g., LLMs). In this work, not only do we show state-of-the-art performances on the Atari100k benchmark, but we also extend transformer-based world models to the continuous domain, where we also report a significant improvement when using the proposed approach. In this paper, we show that Masked Generative Prior improves the sequence modeling capabilities of autoregressive transformers, which touches almost every sub-field of Generative AI.

Chapter 6: Coordination in Multi-Agent RL. The final chapter addresses the role of model-based reinforcement learning in multi-agent coordination. The Stateful Active Facilitator framework demonstrates how agents can achieve cooperation in heterogeneous environments, validating an inductive bias inspired from the Global Workspace Theory [56], a neuroscientific framework about consciousness. By addressing challenges such as sequence modeling, temporal consistency, and multi-agent coordination, this thesis contributes to the broader goal of creating AI systems capable of learning, adapting, and generalising across diverse tasks.

PRELIMINARIES

This chapter provides a mathematically grounded overview of fundamental concepts and methods relevant to the work presented in subsequent chapters. It begins by describing generative modeling frameworks, such as VAEs[14] and VQ-VAEs[57]. Then, we introduce key ideas in video prediction, where latent-variable models are used to forecast future frames from past observations. We then present Slot Attention, which applies attention-based updates to learn object-centric representations. Next, we explore Transformers, covering their general self-attention mechanism and detailing two paradigms: masking bidirectional Transformers and autoregressive Transformers. Finally, we discuss world models and policy learning, emphasizing why the actor-critic approach is well-suited for model-based reinforcement learning (MBRL).

1. VARIATIONAL AUTOENCODERS (VAES)

A *Variational Autoencoder* (VAE)[14] is a powerful generative model that extends the classic autoencoder by introducing a probabilistic latent space. Rather than mapping each input $\mathbf{x}$ (e.g., images) to a single deterministic code, a VAE's encoder learns to produce a *probability distribution* $q_\phi(\mathbf{z} \mid \mathbf{x})$ over a low-dimensional latent variable $\mathbf{z}$. Intuitively, this reflects uncertainty in how complex visual scenes can be represented compactly and enables both reconstruction of inputs and the ability to sample novel scenes from the latent space.

Formally, a VAE comprises three components:

1. A **prior** $p(\mathbf{z})$, typically chosen as an isotropic Gaussian $\mathcal{N}(0, I)$, which imposes a simple, continuous structure on the latent space.

2. An **encoder** (approximate posterior) $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\big(\mathbf{z}; \mu_\phi(\mathbf{x}), \mathrm{diag}(\sigma^2_\phi(\mathbf{x}))\big)$, parameterized by network weights ϕ. It maps each input $\mathbf{x}$ to a mean vector $\mu_\phi(\mathbf{x})$ and standard deviation $\sigma_\phi(\mathbf{x})$.

3. A **decoder** (likelihood) $p_\theta(\mathbf{x} \mid \mathbf{z})$, parameterized by θ, which reconstructs inputs $\mathbf{x}$ from latent $\mathbf{z}$.

Training maximises the Evidence Lower Bound (ELBO) on the log-likelihood of the data:

$$\mathrm{ELBO}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\big).$$

Here, the expected log-likelihood serves as a reconstruction objective that encourages the decoder to faithfully reproduce $\mathbf{x}$ from $\mathbf{z}$. The second term, the Kullback–Leibler divergence between the approximate posterior and the prior, acts as a regularizer

that forces encoded distributions to remain close to $\mathcal{N}(0, I)$. This prevents overfitting and ensures that the latent space is smooth and continuous, enabling meaningful interpolation and sampling.

Because sampling from $q_\phi(\mathbf{z} \mid \mathbf{x})$ would break gradient flow, VAEs employ the *reparameterization trick*:

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

which isolates stochasticity in ϵ, allowing gradients to propagate through μ_ϕ and σ_ϕ.

In video prediction, VAEs are used to extract representations from single frames, encoding each frame $\mathbf{x}_t$ into a latent state $\mathbf{z}_t$ that are then fed to sequential models by defining a transition prior $p(\mathbf{z}_{t+1} \mid \mathbf{z}_{1:t})$. At inference, future latents can be sampled autoregressively and decoded to produce predicted frames. This framework underlies the dissertation's contributions in Chapter 1, which introduces a disentangled VAE variant to isolate independent factors of variation for more interpretable and robust prediction.

2. VECTOR QUANTIZED VARIATIONAL AUTOENCODERS (VQ-VAES)

While VAEs model inputs using continuous latent variables, many visual phenomena—such as distinct object appearances, textures, and categorical scene elements—are naturally discrete. Vector Quantized VAEs (VQ-VAEs) explicitly capture this discreteness by mapping encoder outputs onto a finite, learned vocabulary of embedding vectors, or *codebook* entries. A VQ-VAE comprises three modules:

1. **Encoder** $e_\phi(\mathbf{x})$ produces a dense feature map $\mathbf{E} = \{e_i\}_{i=1}^N \subset \mathbb{R}^d$.
2. **Codebook** $\mathbf{C} = [c_1, \ldots, c_K] \in \mathbb{R}^{d \times K}$ stores K learnable prototype vectors.
3. **Decoder** $p_\theta(\mathbf{x} \mid \mathbf{z})$ reconstructs $\mathbf{x}$ from discrete codes.

Each encoder output e_i is quantized by selecting its nearest codebook entry:

$$z_i = \arg \min_{j \in \{1,\ldots,K\}} \|e_i - c_j\|_2,$$

yielding a discrete *token index* $z_i \in \{1, \ldots, K\}$. The full latent representation $\mathbf{z} = \{z_i\}_{i=1}^N$ thus forms a sequence (or spatial grid) of tokens.

Because quantization is non-differentiable, VQ-VAEs employ a composite loss:

$$\mathcal{L}_{\text{VQ}}(\mathbf{x}) = -\log p_\theta(\mathbf{x} \mid \mathbf{z}) + \alpha \sum_i \|\text{sg}[e_i] - c_{z_i}\|_2^2 + \beta \sum_i \|e_i - \text{sg}[c_{z_i}]\|_2^2.$$

The reconstruction term encourages accurate decoding, the *codebook loss* (weighted by α) updates prototype vectors toward encoder outputs, and the *commitment loss* (weighted by β) aligns encoder outputs to their selected prototypes. The stop-gradient

operator sg[·] blocks gradients as needed to decouple updates. Discrete latents confer multiple benefits over continuous VAEs:

- *Semantic compactness*: Repeated patterns (e.g., object parts) map to the same code, yielding efficient compression.
- *Interpretability*: Each token often corresponds to a distinct semantic concept.
- *Sequence modeling compatibility*: Tokens feed directly into transformer-based predictors, facilitating long-range forecasting with autoregressive or masked generative models.
- *Latency*: Using discrete tokens natively reduces the dimensionality of the latent representation, often resulting in lower latency.

In video prediction, VQ-VAEs compress each frame into a grid of discrete tokens, drastically reducing spatial dimensionality. Subsequent transformer models then predict future token sequences, after which tokens are decoded back to pixel space. This two-step tokenization-and-prediction paradigm enables scalable, high-fidelity long-horizon prediction. As a result, it underlies the transformer-based world models in Chapter 5 and the video prediction model for nowcasting in Chapter 3.

3. VIDEO PREDICTION

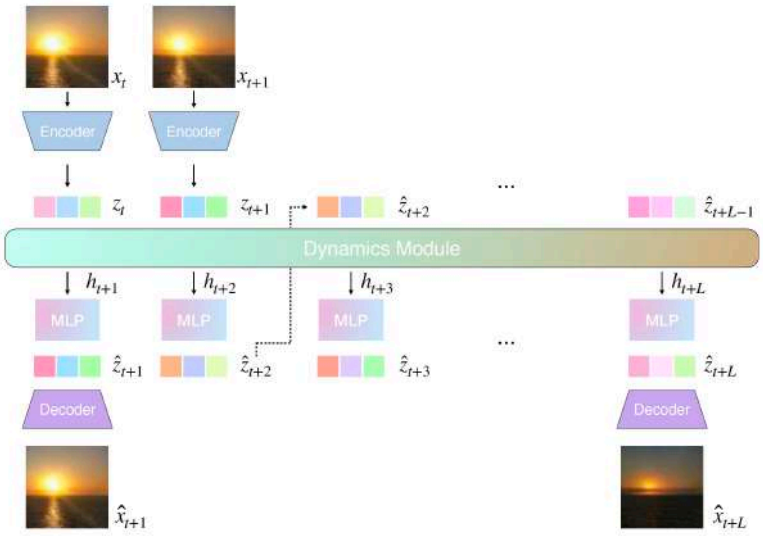


Figure 1: This diagram shows the architecture backbone of a video prediction model. Images $x_{1:T}$ are encoded into latent representations $z_{1:T}$ that are used to predict future states using a latent dynamics module. The predicted latents are then decoded to output future frames.

Video prediction aims to endow artificial systems with the ability to anticipate how a visual scene will evolve. This capability is foundational for safety-critical applications such as autonomous driving [58] (forecasting pedestrian and vehicle trajectories), robotic manipulation [50] (planning in dynamic environments), and weather forecasting [59] (simulating evolving cloud patterns). Historically, pre-deep-learning methods relied on handcrafted optical flow and linear motion extrapolation, which lacked robustness to complex object interactions, occlusions, and high-dimensional pixel distributions. As shown in Fig. 1, in this work, we use a backbone architecture that includes an autoencoder to learn representations z_t and a dynamics module to predict the future representations z_{t+1}. Given an observed clip of T_o frames $\mathbf{X}_{1:T_o} \in \mathbb{R}^{T_o \times H \times W \times C}$, video prediction models learn the conditional distribution

$$P\big(\mathbf{Y}_{T_o+1:T_o+L} \mid \mathbf{X}_{1:T_o}\big),$$

where $\mathbf{Y}_{T_o+1:T_o+L}$ denotes the next L frames. A predictor F outputs $\hat{\mathbf{Y}}_{T_o+1:T_o+L} = F(\mathbf{X}_{1:T_o})$, optimised to minimise a combination of reconstruction, perceptual, and regularization losses [60].

Continuous vs. Discrete Latent Representations. Since learning a dynamics transition at the pixel level is unfeasible, the de-facto approach has become compressing each frame $\mathbf{x}_t$ into a latent representation $\mathbf{z}_t = f_{\text{enc}}(\mathbf{x}_t)$. Within the video prediction literature, two kinds of representations have become the standard; here, we discuss them and analyse the advantages and disadvantages of both.

Continuous Representations. Within the video prediction literature, models like ConvLSTM [61], PredRNN [62] and SimVP [63] have distinguished themselves for their performance and efficiency. They encode each frame into a dense vector $\mathbf{z}_t \in \mathbb{R}^d$ and propagate temporal dynamics via lightweight convolutional or recurrent modules. More specifically, latent representations $\mathbf{z}_t \in \mathbb{R}^d$ evolve via continuous dynamics modules, learning a deterministic mapping

$$\mathbf{z}_{t+1} = g(\mathbf{z}_t).$$

Usually, these models are trained using pixel-level objectives, such as a combination of reconstruction and KL divergence loss:

$$\mathcal{L} = \frac{1}{L} \sum_t \|\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}\|_2^2 + \lambda D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z})).$$

Because operations remain sequential, inference throughput is low, making continuous models hard to use for real-time applications. However, their continuous embeddings can capture highly multimodal futures, often producing better predictions under uncertainty. Continuous models therefore prioritise representation capacity and generative fidelity at the cost of real-time performance.

Discrete Representations. Although continuous latents have high represen-

tation capabilities, lately, most of the video prediction community has started using discrete representations. Usually, latents are quantized into discrete codes via a VQ-VAE [57] encoder. Future codes are then predicted autoregressively by autoregressive transformers [5]:

$$P(\mathbf{z}_{T_o+1:T_o+L} \mid \mathbf{z}_{1:T_o}) = \prod_{t=T_o+1}^{T_o+L} P(\mathbf{z}_t \mid \mathbf{z}_{<t}),$$

trained with cross-entropy loss on codebook indices. While continuos approaches predict one full representation per forward pass, token-based autoregressive approaches predict one token at the time, requiring L forward passes to build the full representation used to decode the next frame. This token-based approach excels at capturing long-range dependencies but requires larger model capacity. Token-based architectures like VideoGPT [60] and MAGVIT[64] excel at modeling long-range dependencies and multimodal outcomes, yielding sharper, more diverse samples. Most importantly, their architecture follows the LLM paradigm, which uses autoregressive transformers for sequence modeling. As a result, this line of research has been advancing rapidly and seems to have higher potential than the one using continuous representations, especially because it has shown higher scalability. The tradeoff lies in inference latency: autoregressive decoding over thousands of tokens can be computationally intensive and typically has lower throughput (tokens/sec) compared to parallelizable methods, however it enables modeling very complex distributions.

Connections to Thesis Chapters. The continuous latent formulation underlies Chapter 1, which develops disentangled VAEs for interpretable latent factorisation. The discrete token-based framework informs Chapter 2, enabling object-centric token modeling for robust multi-object tracking. Chapters 3 and 4 leverage a VQGAN [65] to extract discrete representations from precipitation maps and use an autoregressive transformer to perform sequence modeling of the learned tokens. Chapter 5 builds on transformer-based world models in discrete latent space to support long-horizon planning and control in model-based reinforcement learning.

4. SLOT ATTENTION

As showed in Fig. 2, Slot Attention is a neural module designed to learn object-centric [66] representations from raw perceptual inputs by decomposing a scene into a fixed set of "slots," each intended to bind to one object or coherent region. This approach addresses a fundamental limitation of conventional deep networks, whose distributed representations entangle multiple entities and fail to capture the compositional structure of natural scenes. By enforcing a competitive, permutation-equivariant clustering over learned features, Slot Attention yields interpretable, modular latent variables that improve generalization, sample efficiency, and downstream reasoning.

Given an image encoded into a set of feature vectors $\mathbf{F} \in \mathbb{R}^{N \times D}$ (where N tokens represent spatial patches and D their embedding dimension), Slot Attention initializes

K learnable slot vectors $\{\mathbf{s}_k\}_{k=1}^{K} \in \mathbb{R}^D$. Over T iterative refinement steps, each slot competes to explain parts of the input via scaled dot-product attention:

$$\alpha_{ik} = (\mathbf{F}_i \mathbf{W}_Q)(\mathbf{s}_k \mathbf{W}_K)^\top, \quad w_{ik} = \frac{\exp(\alpha_{ik})}{\sum_{k'} \exp(\alpha_{ik'})}.$$

Where $\mathbf{W}_Q \in \mathbb{R}^{D\times K}$ and $\mathbf{W}_K \in \mathbb{R}^{D\times N}$ are the query and key transformation matrices. Slots update by aggregating weighted features:

$$\mathbf{u}_k = \sum_{i=1}^{N} w_{ik}\, \mathbf{F}_i, \quad \mathbf{s}_k \leftarrow \mathrm{GRU}(\mathbf{s}_k, \mathbf{u}_k) + \mathrm{MLP}(\mathbf{s}_k).$$

Normalization of attention across slots induces competition, ensuring each slot specializes in a distinct object or region. In unsupervised object discovery, slots

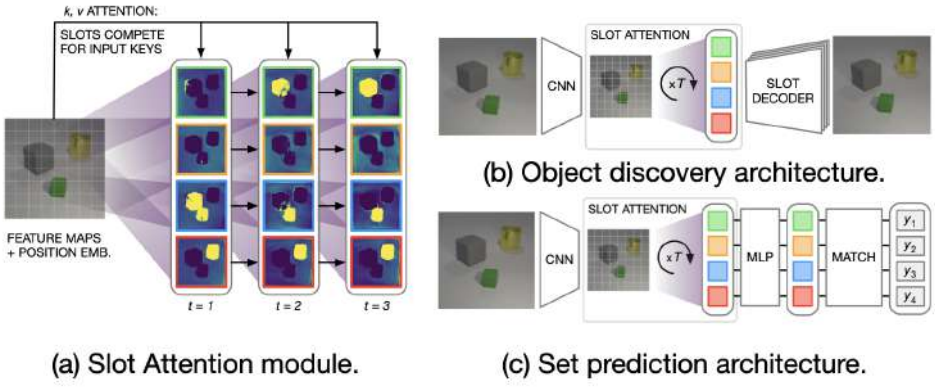


Figure 2: Slot attention backbone module, object discovery and set prediction architectures [66].

are decoded independently via a spatial broadcast decoder [67] to reconstruct the input image, and training minimises reconstruction loss alone. In supervised set prediction, slot outputs feed a shared MLP classifier and are matched to ground-truth object properties via the Hungarian algorithm, optimising cross-entropy losses for each property. Slot Attention yields permutation-invariant, disentangled object representations that generalise to novel object counts and compositions. It is computationally efficient—requiring only a few attention iterations—yet sensitive to the choice of slot number K and iterations T. Background segmentation is implicit and may require additional mechanisms for complex scenes. Slot Attention forms the foundation of Chapter 2, where we employ object-centric representations for video sequences for robust object tracking and temporally consistent predictions.

5. TRANSFORMERS

Transformer networks have revolutionized sequence and set modeling, marking a significant departure from previous recurrent and convolutional architectures [5]. This shift is primarily driven by the introduction of the multi-head self-attention mechanism, which enables the model to process all input positions simultaneously, a stark contrast to the sequential processing of recurrent neural networks. This parallelism facilitates the efficient capture of long-range dependencies, a crucial aspect for understanding context in sequential data. For instance, in a long sentence, the relationship between words far apart from each other can be directly modeled, overcoming the limitations of recurrent models that often struggle with vanishing or exploding gradients when processing long sequences [62]. Furthermore, the Transformer architecture supports permutation-invariant set processing and scales efficiently to very large models and datasets, making it a foundational backbone for modern language [68], vision [69], and multimodal systems [70]. The motivation behind the Transformer architecture stems from the limitations of recurrent and convolutional neural networks in sequence transduction tasks. Recurrent models, while effective in capturing sequential dependencies, process input tokens sequentially, hindering parallelization and making it challenging to capture long-range dependencies due to the inherent sequential computation. Convolutional networks, on the other hand, can process parts of the input in parallel but require stacking multiple layers to capture long-range dependencies, and the receptive field grows linearly with the number of layers. Transformer architectures address these limitations by leveraging the attention mechanism, allowing for parallel processing and direct modeling of relationships between any two positions in the input sequence. This enables the model to efficiently capture both short-range and long-range dependencies, leading to improved performance in several sequence transduction tasks (e.g., machine translation [71], question answering [72], text summarization [73], etc.). The Transformer architecture, as showed in Fig. 3, is primarily based on the attention mechanism. Let's consider a sequence represented by a matrix $\mathbf{X} \in \mathbb{R}^{n \times d_{\text{model}}}$, which represents a sequence length n and feature dimension d_{model}. The self-attention mechanism computes a weighted sum of the input features, where the weights are determined by the relationships between different positions in the input sequence.

First, the input $\mathbf{X}$ is linearly transformed into three matrices: queries $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$, keys $\mathbf{K} \in \mathbb{R}^{n \times d_k}$, and values $\mathbf{V} \in \mathbb{R}^{n \times d_v}$:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V,$$

where $\mathbf{W}_Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $\mathbf{W}_V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are the query, key, and value transformation matrices. Moreover, d_k is the dimensionality of the queries and keys, and d_v is the dimensionality of the values.

A scaled dot-product of the queries and keys is used to compute the attention weights, which are then normalized by a softmax operation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}.$$

The scaling factor $\sqrt{d_k}$ stabilises gradients during training.

Multi-head attention parallelizes this mechanism by performing the attention calculation H times using different linear projections. A linear transformation computes the final output using the concatenated projected results as input:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}_O,$$

where $\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_Q^i, \mathbf{K}\mathbf{W}_K^i, \mathbf{V}\mathbf{W}_V^i)$ and $\mathbf{W}_O \in \mathbb{R}^{Hd_v \times d_{\text{model}}}$ represents the output transformation matrix.

A Transformer layer typically consists of a multi-head self-attention layer and a position-wise feedforward network (FFN). The FFN applies the same fully connected feedforward network to every position separately, consisting of two linear transformations and a ReLU activation between them:

$$\text{FFN}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W}_1)\mathbf{W}_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ are the parameters of the feedforward network, and d_{ff} is the inner-layer dimensionality.

Layer normalization and residual connections are employed in every layer [5].

The optimization procedure for training Transformer models involves minimizing a loss function that measures the difference between the model's predictions and the ground truth. The choice of the loss function depends on the specific task.

For sequence transduction tasks like machine translation, the model is typically trained using a cross-entropy loss. Given a source sequence $\mathbf{x} = (x_1, \ldots, x_n)$ and a target sequence $\mathbf{y} = (y_1, \ldots, y_m)$, the model learns the conditional probability $p(\mathbf{y}|\mathbf{x})$. The cross-entropy loss is then defined as:

$$\mathcal{L} = -\sum_{i=1}^{m} \log p(y_i|y_{<i}, \mathbf{x}),$$

where $p(y_i|y_{<i}, \mathbf{x})$ is the probability distribution of the i-th target representation given the previous target ones and the source sequence. The model predicts this probability distribution using the Transformer architecture, and the parameters of the model (i.e., the weights $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$, $\mathbf{W}_O$, $\mathbf{W}_1$, and $\mathbf{W}_2$ in each layer) are optimised to minimise this loss.

The motivation behind using cross-entropy loss is to optimise the likelihood of the correct target sequence, given a source sequence. By optimising the cross-entropy loss, the model learns to predict the target sequence with high accuracy. The optimization process is typically performed using gradient-based optimization algorithms, such as Adam [74], which iteratively updates the model parameters based on the gradients of the loss function.

5.1. VISION TRANSFORMERS

Vision Transformers (ViT) adapt the Transformer architecture for image recognition tasks [75]. This adaptation involves patchifying an image and extracting tokens from

these patches as a sequence, as showed in Fig. 3.

Given an input image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, where H is the height, W is the width, and C is the number of channels, N patches of size $P \times P$ can be extracted. All patches are then flattened into vectors of size P^2C. Each flattened patch is projected into an embedding space of dimension d_{model}, resulting in a sequence of patch embeddings:

$$\mathbf{X}_p \in \mathbb{R}^{N \times (P^2 C)}, \quad \mathbf{E} \in \mathbb{R}^{N \times d_{\text{model}}},$$

where $N = \frac{HW}{P^2}$ is the number of patches, and $\mathbf{E}$ is the matrix of embedded patches. To retain spatial information, which is lost during the flattening process, positional

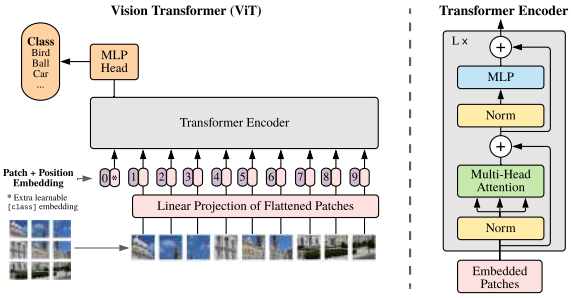


Figure 3: Visualisation of vision transformer architecture.

encodings are added to the patch embeddings. These positional encodings provide the model with information about the location of each patch in the original image. The positional encodings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N \times d_{\text{model}}}$ are added to the patch embeddings:

$$\mathbf{X}_{\text{emb}} = \mathbf{E} + \mathbf{E}_{\text{pos}}.$$

The resulting embeddings are then fed into a standard Transformer encoder, which consists of several multi-head self-attention layers and feedforward networks.

The optimization procedure for ViT models is similar to that of standard Transformers. For image classification, the model is typically trained using a cross-entropy loss. The output of the Transformer encoder is passed through a classification head, which consists of one or more linear layers, to produce the class predictions. The model parameters are optimised to minimise the cross-entropy loss between the predicted class probabilities and the ground truth labels.

5.2. BIDIRECTIONAL TRANSFORMERS

Bidirectional Transformers, such as BERT (Bidirectional Encoder Representations from Transformers) [76], leverage bidirectional context to obtain richer contextual representations. This is achieved through masked language modeling (MLM), a pre-training objective that involves masking certain tokens in the input sequence and training the model to predict the masked tokens.

Considering an input sequence $\mathbf{x} = (x_1, \ldots, x_n)$, the MLM objective involves randomly masking a subset of the tokens in the input sequence. Let $\mathbf{m} = (m_1, \ldots, m_n)$

be a binary mask, where $m_i = 1$ if the i-th token is masked and $m_i = 0$ otherwise. The model is trained to predict the masked tokens given the context of the unmasked tokens.

The optimization objective for MLM is to minimise the negative log-likelihood of the masked tokens:

$$\mathbb{L}_{\text{MLM}} = -\mathbb{E}_{\mathbf{x},\mathbf{m}} \left[\sum_{i=1}^{n} m_i \log p(x_i | \mathbf{x}_{\overline{\mathbf{m}}}) \right],$$

where $\mathbf{x}_{\overline{\mathbf{m}}}$ denotes the unmasked tokens in the input sequence.

Masking is crucial for bidirectional Transformers because it allows the model to leverage both left and right context during training. In contrast to autoregressive models, which can only attend to previous tokens, bidirectional models can attend to both previous and subsequent tokens, leading to richer contextual representations. Masking prevents the model from trivially "seeing" the target token during training, forcing it to rely on contextual information to make predictions.

The intuition behind MLM is to train the model to understand the relationships between different words in a sentence. By predicting masked tokens, the model learns to capture the semantic and syntactic dependencies between words, leading to improved performance on tasks like question answering and text classification.

In Chapter 5, we will see how using a masked bidirectional transformer to learn a generative prior for world models' dynamics modules improves their sequence modeling capabilities.

5.3. AUTOREGRESSIVE TRANSFORMERS

Autoregressive Transformers are designed for sequence generation tasks, where the goal is to predict the next token in a sequence given the previous tokens. These models enforce causality by masking future positions in the self-attention mechanism, ensuring that the prediction for each position only depends on the past context.

Given a discrete sequence $\mathbf{s} = (s_1, \ldots, s_N)$, the joint probability distribution of the sequence is factorized as:

$$p_\theta(\mathbf{s}) = \prod_{i=1}^{N} p_\theta(s_i | s_{<i}),$$

where $p_\theta(s_i | s_{<i})$ represents the conditional probability distribution of the i-th token, given the previous ones $s_{<i} = (s_1, \ldots, s_{i-1})$.

The optimization procedure for autoregressive Transformers involves minimizing the negative log-likelihood of the sequence, which is equivalent to minimizing the cross-entropy loss:

$$\mathbb{L}_{\text{AT}} = -\mathbb{E}_{\mathbf{s}} \left[\sum_{i=1}^{N} \log p_\theta(s_i | s_{<i}) \right].$$

The motivation behind this autoregressive formulation is to learn the underlying

probability distribution of the data, allowing the model to generate sequences in a step-by-step manner. At each step, the model predicts the probability distribution over the possible next tokens, and a token is sampled from this distribution. This process is repeated until the end of the sequence is reached.

The intuition behind autoregressive modeling is to capture the sequential dependencies in the data. By conditioning the prediction of each token on the previous tokens, the model learns to model the order and structure of the sequence. This is particularly important for tasks such as language modeling, where the generated text needs to be coherent and grammatically correct.

In the Transformer architecture, causality is enforced by masking future positions within the attention mechanism. Specifically, attention weights are modified to ensure that each position can exclusively attend to previous positions. This is typically achieved by applying a lower triangular mask to the attention weight matrix before the softmax operation.

Autoregressive Transformers are now successfully used for various sequence generation tasks, including language modeling (e.g., GPT models) [77], audio generation [78], and time-series forecasting [79].

Autoregressive Transformers play a crucial role in Chapters 2, 3, 4, and 5, since we employ them as the dynamics module in the models presented in these chapters.

6. WORLD MODELS

The concept of world models provides a compelling framework for enabling intelligent agents to acquire an understanding of their operational environment and to reason effectively within it. At the core of this paradigm lies the objective of learning a parameterized function, typically denoted as $p_\phi(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{a}_t)$, which aims to capture the underlying dynamics governing the evolution of latent state representations $\mathbf{z}_t$ in response to the agent's actions $\mathbf{a}_t$ [8, 26]. Through the acquisition of such a predictive model, an agent gains the capability to simulate future state transitions and generate hypothetical trajectories, represented as $\mathbf{z}_{t+1}, \mathbf{z}_{t+2}, \ldots$, without necessitating direct interaction with the real-world environment. This approach proves particularly advantageous in scenarios where the collection of real-world data is either prohibitively expensive in terms of resources or entails significant risks. As visually depicted in Fig. 4, the fundamental architectural structure of world models exhibits notable similarities with that of video prediction models. However, a critical distinction arises in the manner in which future latent representations are conditioned. In the context of world models, these representations are explicitly conditioned on the actions undertaken by the agent. Consequently, the problem of sequence modeling transitions into the learning of state transition dynamics, where the evolution of the state is predicted based on the current state and the specific action executed by the agent. Furthermore, the Multi-Layer Perceptron (MLP) head commonly employed in video prediction backbones is adapted in world models to serve as a dynamics prior. To accommodate the episodic nature inherent in many real-world tasks, where episodes can conclude at various points within a sequence, a termination head is typically incorporated to predict the cessation of episodes. Finally, recognising the integral role of rewards in

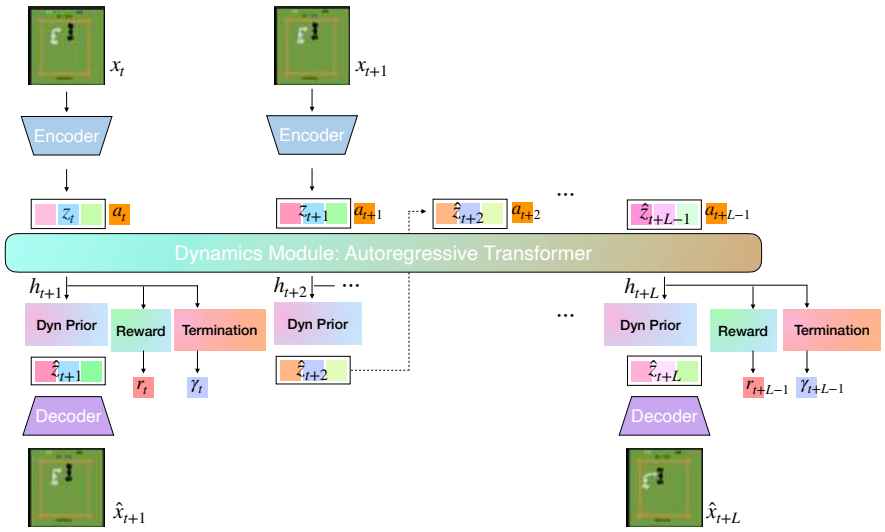


Figure 4: This diagram shows a general world model architecture.

reinforcement learning, where each state transition resulting from an action yields a specific reward, a reward head is included to predict the magnitude of the received reward. The methodology presented herein, which is elaborated upon in detail in Section 6.1, aligns with the well-established principles of model-based reinforcement learning (MBRL) algorithms. These algorithms leverage the learned world model as a crucial component in enhancing the agent's policy through a process often referred to as imagination [26, 80–83]. The learning process typically involves an iterative cycle comprising data acquisition, world model refinement, and policy enhancement facilitated by experiences generated through the world model's simulations. This cyclical process continues until a predetermined number of interactions with the real environment has been achieved. The iterative learning process can be systematically outlined as follows:

S1) **Real-World Data Acquisition:** The agent's current policy is executed within the actual environment for a specified duration, and the resulting data points, encompassing observations, actions, rewards, and continuation flags, are appended to a replay buffer for subsequent use.

S2) **World Model Parameter Update:** The parameters of the world model are updated through training on sequences of trajectories sampled from the replay buffer. The main objective of this step is to improve the model's predictive accuracy with respect to future states, rewards, and the termination of episodes.

S3) **Policy Refinement via Imagined Experiences:** Using experiences generated by the world model, the agent's policy is trained without needing real-world interactions. The initial states for these imagined trajectories are

typically sampled from the replay buffer, enabling the agent to learn and adapt based on these synthetic rollouts.

At each discrete time step t, a data point gathered from the environment typically comprises an observation $\mathbf{o}_t$, an action $\mathbf{a}_t$ executed by the agent, a reward r_t received as a consequence of the action, and a continuation flag c_t. This continuation flag is a binary indicator that represents whether the current episode has ended at time t. The defined replay buffer employs a first-in-first-out (FIFO) structure, which facilitates the sampling operation of temporally contiguous sequences. This is particularly useful for training sequence-based world models and for providing coherent starting points for the imagination process.

Section 6.1 explains the architectural design and training objectives used by the world model, referred to as STORM in this work. Furthermore, Section 6.2 provides a comprehensive elaboration on the process of imagination and the training methodology utilised to refine the agent's policy by effectively utilizing the predictive capabilities of the learned world model.

6.1. WORLD MODEL LEARNING

Given image observations $\mathbf{o}_t$ representing the environment's state, modeling the state-transition dynamics of the environment from pixel-level inputs presents significant computational demands and can be susceptible to the accumulation of errors [26, 81–86]. To address these challenges, a Variational Autoencoder (VAE) [14] is commonly employed. Consistent with prior research endeavors in this domain [26, 81], the latent distribution $\mathcal{Z}_t$ is typically configured as a categorical distribution consisting of 32 distinct categories, with each category further subdivided into 32 discrete classes. The encoder network (q_ϕ) and the decoder network (p_ϕ) are commonly implemented using Convolutional Neural Networks (CNNs) [87], which are well-suited for processing image data. Subsequently, a latent variable $\mathbf{z}_t$ is sampled from the distribution $\mathcal{Z}_t$ to serve as a compact representation of the original observation $\mathbf{o}_t$. To enable the propagation of gradients through the discrete sampling process, which is inherently non-differentiable, the straight-through gradients trick [81, 88] is typically applied. The formulation of this VAE is provided below, where it serves to transform the observation $\mathbf{o}_t$ into a low-dimensional latent stochastic categorical distribution $\mathcal{Z}_t$.

$$\begin{aligned} &\text{Image encoder:} \quad \mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{o}_t) = \mathcal{Z}_t \\ &\text{Image decoder:} \quad \hat{\mathbf{o}}_t = p_\phi(\mathbf{z}_t). \end{aligned}$$

The sampled representation $\mathbf{z}_t$ and the action $\mathbf{a}_t$ are typically combined into an embedding $\mathbf{e}_t$ through the use of an MLP and a concatenation operation. This operation, denoted as m_ϕ, prepares the inputs for the subsequent sequence modeling stage. The sequence model f_ϕ then takes a sequence of these combined embeddings $\mathbf{e}_t$ as input and produces a corresponding sequence of hidden states $\mathbf{h}_t$. A common choice for the sequence model architecture in recent work is a GPT-like Transformer structure [89]. The self-attention mechanisms within the Transformer are typically configured with a subsequent mask, ensuring that the embedding at any given time step t can only attend to the sequence of embeddings up to and including that

time step (i.e., $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_t\}$). Finally, three distinct MLP heads, denoted as g_ϕ^D, g_ϕ^R, and g_ϕ^C, are employed. These heads take the hidden state $\mathbf{h}_t$ as input and are responsible for predicting the parameters of the next latent distribution $\hat{\mathcal{Z}}_{t+1}$, the immediate reward $\hat{r}_t$, and the continuation flag $\hat{c}_t$, respectively. To sum up, world models can be formalized as follows:

$$\begin{aligned}
&\text{Action mixer:} && \mathbf{e}_t = m_\phi(\mathbf{z}_t, \mathbf{a}_t) \\
&\text{Sequence model:} && \mathbf{h}_{1:T} = f_\phi(\mathbf{e}_{1:T}) \\
&\text{Dynamics predictor:} && \hat{\mathcal{Z}}_{t+1} \sim g_\phi^D(\hat{\mathbf{z}}_{t+1}|\mathbf{h}_t) \\
&\text{Reward predictor:} && \hat{r}_t = g_\phi^R(\mathbf{h}_t) \\
&\text{Continuation predictor:} && \hat{c}_t = g_\phi^C(\mathbf{h}_t).
\end{aligned}$$

World models are typically trained using a self-supervised approach, with the primary objective of minimizing a comprehensive loss function that aggregates several individual loss components. The final objective function is commonly formulated as follows, with hyperparameters $\beta_1 = 0.5$ and $\beta_2 = 0.1$. Within this formulation, B and T typically represent the batch size and the length of the sampled trajectories, respectively.

$$\mathcal{L}(\phi) = \frac{1}{BT} \sum_{n=1}^{B} \sum_{t=1}^{T} \Big[\mathcal{L}_t^{\text{rec}}(\phi) + \mathcal{L}_t^{\text{rew}}(\phi) + \mathcal{L}_t^{\text{con}}(\phi) + \beta_1 \mathcal{L}_t^{\text{dyn}}(\phi) + \beta_2 \mathcal{L}_t^{\text{rep}}(\phi) \Big].$$

$\mathcal{L}_t^{\text{rec}}(\phi)$ is the reconstruction loss, which quantifies the discrepancy between the ground truth image $\mathbf{o}_t$ and the reconstructed image $\hat{\mathbf{o}}_t$ produced by the decoder. $\mathcal{L}_t^{\text{rew}}(\phi)$ denotes the reward prediction loss, which measures the error in the world model's prediction of the reward r_t. Similarly, $\mathcal{L}_t^{\text{con}}(\phi)$ represents the continuation prediction loss, which evaluates the accuracy of the predicted continuation flag $\hat{c}_t$ in relation to the true continuation flag c_t. These individual loss terms are often defined as follows:

$$\begin{aligned}
\mathcal{L}_t^{\text{rec}}(\phi) &= ||\hat{\mathbf{o}}_t - \mathbf{o}_t||_2 \\
\mathcal{L}_t^{\text{rew}}(\phi) &= \mathcal{L}^{\text{sym}}(\hat{r}_t, r_t) \\
\mathcal{L}_t^{\text{con}}(\phi) &= c_t \log \hat{c}_t + (1 - c_t) \log(1 - \hat{c}_t).
\end{aligned}$$

In the reward loss, $\mathcal{L}^{\text{sym}}$ typically refers to the symlog two-hot loss, as in prior work such as [26]. This specific loss function is often employed to transform the reward regression problem into a classification problem, which can help ensure more consistent loss scaling across different environments that may exhibit varying ranges of reward values.

The losses $\mathcal{L}_t^{\text{dyn}}(\phi)$ and $\mathcal{L}_t^{\text{rep}}(\phi)$, which are defined below, are both formulated as Kullback–Leibler (KL) divergences. However, they differ in terms of how gradients are backpropagated and the weighting applied to each term. The dynamics

loss $\mathcal{L}_t^{\text{dyn}}(\phi)$ serves to guide the dynamics model in accurately learning the state-transition dynamics of the latent space. Conversely, the representation loss $\mathcal{L}_t^{\text{rep}}(\phi)$ improves the encoder's capabilities to incorporate dynamics information within the extracted representations. The stop-gradient operation, denoted as $\text{sg}(\cdot)$, is utilised to prevent gradients from flowing backward through the specified variables during the optimization process.

$$\mathcal{L}_t^{\text{dyn}}(\phi) = \max\big(1, \text{KL}\big[\text{sg}(q_\phi(\mathbf{z}_{t+1}|\mathbf{o}_{t+1})) \;||\; g_\phi^D(\hat{\mathbf{z}}_{t+1}|\mathbf{h}_t)\big]\big)$$
$$\mathcal{L}_t^{\text{rep}}(\phi) = \max\big(1, \text{KL}\big[q_\phi(\mathbf{z}_{t+1}|\mathbf{o}_{t+1}) \;||\; \text{sg}(g_\phi^D(\hat{\mathbf{z}}_{t+1}|\mathbf{h}_t))\big]\big)$$

6.2. AGENT LEARNING

Within this framework, the agent's learning process happens in imagination. The trained world model generates trajectories that allow the agent to learn meaningful policies without interacting with the real world. To align imagined trajectories with real ones, a few context steps are used to condition the generation of the imagined steps. For the initial observation within this context, the posterior distribution over the latent space, $\mathcal{Z}_t$, is computed using the encoder. During the subsequent inference steps within the imagined trajectory, instead of sampling from the posterior $\mathcal{Z}_t$, the latent variable $\mathbf{z}_t$ is sampled using the dynamics prior $\hat{\mathcal{Z}}_t$. To enhance the efficiency of the inference process, particularly within the Transformer architecture, the Key-Value (KV) cache technique [37] is often employed. This technique optimises the computation of attention mechanisms, which are crucial for the efficient generation of autoregressive sequences.

The agent's state at each time step t, denoted as $\mathbf{s}_t$, is typically constructed by concatenating the sampled latent variable $\mathbf{z}_t$ with the hidden state $\mathbf{h}_t$ produced by the sequence model:

$$\begin{aligned}
&\text{State:} && \mathbf{s}_t = [\mathbf{z}_t, \mathbf{h}_t] \\
&\text{Critic:} && V_\psi(\mathbf{s}_t) \approx \mathbb{E}_{\pi_\theta, p_\phi}\Big[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\Big] \\
&\text{Actor:} && \mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t).
\end{aligned}$$

The actor-critic learning framework and associated settings, as established in prior work such as DreamerV3 [26], are frequently adopted in this context. The complete loss function utilised for training the actor (policy π_θ) and the critic (V_ψ) is generally described below. In these equations, $\hat{r}_t$ represents the predicted reward at time t within the imagined trajectory, and $\hat{c}_t$ represents the predicted continuation flag for the same time step. η and L denote the entropy regularization term and the length of the imagination horizon (i.e., the number of steps in the imagined trajectories), respectively. The batch size used during the agent training process is denoted by B.

$$\mathcal{L}(\theta) = \frac{1}{BL}\sum_{n=1}^{B}\sum_{t=1}^{L}\left[-\mathrm{sg}\left(\frac{G_t^\lambda - V_\psi(\mathbf{s}_t)}{\max(1,S)}\right)\ln \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) - \eta H\big(\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\big)\right]$$

$$\mathcal{L}(\psi) = \frac{1}{BL}\sum_{n=1}^{B}\sum_{t=1}^{L}\left[\Big(V_\psi(\mathbf{s}_t) - \mathrm{sg}\big(G_t^\lambda\big)\Big)^2 + \Big(V_\psi(\mathbf{s}_t) - \mathrm{sg}\big(V_{\psi^{\mathrm{EMA}}}(\mathbf{s}_t)\big)\Big)^2\right]$$

The actor loss aims to optimise the agent's policy by maximizing the expected cumulative reward obtained in the imagined trajectories. Simultaneously, an entropy regularization term is included to encourage exploration by the agent, preventing premature convergence to suboptimal deterministic policies. The critic loss aims to improve the accuracy of the value function by minimizing the squared error between the predicted value $V_\psi(\mathbf{s}_t)$ and a target value derived from the λ-return, G_t^λ.

The λ-return G_t^λ [26, 80] provides a mechanism for temporally consistent credit assignment, effectively blending the benefits of short-term and long-term return estimates. It is recursively defined as follows:

$$G_t^\lambda \doteq \hat{r}_t + \gamma\hat{c}_t\Big[(1-\lambda)V_\psi(\mathbf{s}_{t+1}) + \lambda G_{t+1}^\lambda\Big]$$
$$G_L^\lambda \doteq V_\psi(\mathbf{s}_L).$$

In this formulation, γ represents the discount factor, which determines the importance of future rewards relative to immediate rewards, and λ is a parameter that controls the trade-off between relying on the value function for bootstrapping and using more extensive, multi-step return estimates.

The normalization ratio S used within the actor loss, as shown below, is typically defined as the difference between the 95th and 5th percentiles of the λ-return G_t^λ across the entire batch of imagined trajectories [26]. This normalization step helps to stabilise the training of the actor by scaling the gradients based on the distribution of the returns, thereby mitigating issues arising from large variations in return magnitudes.

$$S = \mathrm{percentile}(G_t^\lambda, 95) - \mathrm{percentile}(G_t^\lambda, 5).$$

To further enhance the stability of the value function training and to prevent overfitting, an exponential moving average (EMA) of the critic parameters ψ is often maintained. The EMA is updated according to the following rule, where ψ_t is the critic parameter at time step t, σ is the decay rate that governs the influence of the previous EMA value, and $\psi_{t+1}^{\mathrm{EMA}}$ denotes the updated EMA parameters at time step $t+1$.

$$\psi_{t+1}^{\mathrm{EMA}} = \sigma\psi_t^{\mathrm{EMA}} + (1-\sigma)\psi_t.$$

7. CONCLUSION

This chapter has provided a comprehensive mathematical foundation for the key concepts and methodologies that underpin the research presented in this dissertation. We began by examining generative modeling frameworks, establishing the theoretical basis for both continuous (VAEs) and discrete (VQ-VAEs) latent representations. These representation learning paradigms form the foundation for encoding high-dimensional visual data into compact, semantically meaningful latent spaces. We then explored video prediction, highlighting the fundamental trade-offs between continuous and discrete latent approaches. While continuous representations offer rich modeling capacity for multimodal futures, discrete tokenization enables scalable transformer-based architectures that have demonstrated superior long-range dependency modeling and alignment with the rapidly advancing large language model paradigm. The introduction of Slot Attention provided crucial insights into object-centric representation learning, demonstrating how attention mechanisms can decompose complex scenes into interpretable, permutation-invariant object representations. This foundation proves essential for handling multi-object scenarios and achieving compositional generalization. Our examination of Transformer architectures—spanning self-attention mechanisms, vision transformers, bidirectional masked models, and autoregressive variants—established the sequence modeling capabilities that enable effective temporal dynamics learning. These architectures serve as the backbone for the predictive models developed throughout this thesis. Finally, we presented world models as a unifying framework that bridges video prediction and reinforcement learning. By learning state transition dynamics in latent space and enabling policy optimization through imagination, world models provide a principled approach to sample-efficient learning in complex environments. The theoretical foundations laid out in this chapter directly inform the contributions of the subsequent chapters: Chapter 1 builds upon disentangled VAE representations, Chapter 2 leverages object-centric modeling through Slot Attention, Chapters 3 and 4 employ discrete tokenization for precipitation forecasting, and Chapter 5 advances transformer-based world models for reinforcement learning. Together, these preliminaries establish the mathematical and conceptual framework necessary for understanding the novel methodologies and empirical findings that follow.

1

α-TCVAE: ON THE RELATIONSHIP BETWEEN DISENTANGLEMENT AND DIVERSITY

Published at the Twelfth International Conference on Learning Representations (ICLR 2024)

Cristian Meo
Delft University of Technology, NL
c.meo@tudelft.nl

Louis Mahon
University of Edinburgh, UK

Anirudh Goyal
Google DeepMind, UK

Justin Dauwels
Delft University of Technology, NL

1

Understanding and developing optimal representations has long been foundational in machine learning (ML). While disentangled representations have shown promise in generative modeling and representation learning, their downstream usefulness remains debated. Recent studies re-defined disentanglement through a formal connection to symmetries, emphasizing the ability to reduce latent domains (i.e., ML problem spaces) and consequently enhance data efficiency and generative capabilities. However, from an information theory viewpoint, assigning a complex attribute (i.e., features) to a specific latent variable may be infeasible, limiting the applicability of disentangled representations to simple datasets. In this work, we introduce α-TCVAE, a variational autoencoder optimised using a novel total correlation (TC) lower bound that maximises disentanglement and latent variables informativeness. The proposed TC bound is grounded in information theory constructs, generalises the β-VAE lower bound, and can be reduced to a convex combination of the known variational information bottleneck (VIB) and conditional entropy bottleneck (CEB) terms. Moreover, we present quantitative analyses and correlation studies that support the idea that smaller latent domains (i.e., disentangled representations) lead to better generative capabilities and diversity. Additionally, we perform downstream task experiments from both representation and RL domains to assess our questions from a broader ML perspective. Our results demonstrate that α-TCVAE consistently learns more disentangled representations than baselines and generates more diverse observations without sacrificing visual fidelity. Notably, α-TCVAE exhibits marked improvements on MPI3D-Real, the most realistic disentangled dataset in our study, confirming its ability to represent complex datasets when maximizing the informativeness of individual variables. Finally, testing the proposed model off-the-shelf on a state-of-the-art model-based RL agent, Director, significantly shows α-TCVAE downstream usefulness on the loconav Ant Maze task.

1.1. INTRODUCTION

The efficacy of machine learning (ML) algorithms is intrinsically tied to the quality of data representation [90]. Such representations are useful not only for standard downstream tasks such as supervised learning [91] and reinforcement learning (RL) [92], but also for tasks such as transfer learning [93] and zero-shot learning [94]. Unsupervised representation learning aims to identify semantically meaningful representations of data without supervision, by capturing the generative factors of variations that describe the structure of the data [95, 96]. According to [90], disentanglement learning holds the key to understanding the world from observations, generalising knowledge across different tasks and domains while learning and generating compositional representations [97, 98].

Problem Formulation. The goal of disentanglement learning is to identify a set of independent generative factors $\boldsymbol{z}$ that give rise to the observations $\boldsymbol{x}$ via $p(\boldsymbol{x}|\boldsymbol{z})$. However, from an information theory perspective, the amount of information retained by every latent variable may be insufficient to represent realistic generative factors [99, 100], limiting the applicability of disentangled representations to simple problems.

What is more, [101] recently introduced the Vendi score, a new metric for gauging generative diversity, showing that entangled generative models, such as the Very Deep VAE [102], consistently produce samples with less diversity compared to ground truth. This is indicative of their limited representational and generative prowess. In contrast, [103, 104] re-defined disentangled representations through the lens of symmetries, linking disentanglement to computational problem spaces (e.g., disentangled representations inherently reduce the problem space [105]), suggesting that disentangled models should be able to explore and traverse the latent space more efficiently, leading to enhanced generative diversity.

Previous Work. Most existing disentangled models optimise lower bounds that only impose an information bottleneck on the latent variables, and while this can result in factorized representations [97], it does not directly optimise latent variable informativeness [100]. As a result, while several approaches have been proposed to learn disentangled representations by optimising different bounds [98, 106], imposing sparsity priors [107], or isolating source of variance [108], none of the proposed models successfully learned disentangled representations of realistic datasets. Moreover, to the best of our knowledge, no systematic and quantitative analyses have been proposed to assess to what extent disentanglement and generative diversity [101] are correlated.

Proposed method. In this work, we propose α-TCVAE, a VAE optimised using a novel convex lower bound of the joint total correlation (TC) between the learned latent representation and the input data. The proposed bound, through a convex combination of the variational information bottleneck (VIB) [91] and the conditional entropy bottleneck (CEB) [109], maximises the average latent variable informativeness, improving both representational and generative capabilities. The effectiveness of α-TCVAE is especially prominent in the MPI3D-Real Dataset [110], the most realistic dataset in our study that is compositionally built upon distinct generative factors. Figure 1.1 illustrates a comparison of the latent traversals between α-TCVAE, Factor-VAE and β-VAE, showing that α-TCVAE leads to the best visual fidelity and generative diversity (i.e., higher Vendi Score). Interestingly, the proposed TC bound is grounded in information theory constructs, generalises the β-VAE [97] lower bound, and can be reduced to a convex combination of the known variational information bottleneck (VIB) [91] and conditional entropy bottleneck (CEB) [109] terms.

Experimental Evaluation In order to determine the effectiveness of α-TCVAE and the downstream usefulness of the learned representations, we measure the diversity and quality of generated images and disentanglement of its latent representations. Then, we perform a correlation study between the considered downstream scores across all models, analyzing how generative diversity and disentanglement are related across different datasets. This analysis substantiates our claim that disentanglement leads to improved diversity. Finally, we conduct experiments to assess the downstream usefulness of the proposed method from a broader ML perspective. Notably, the

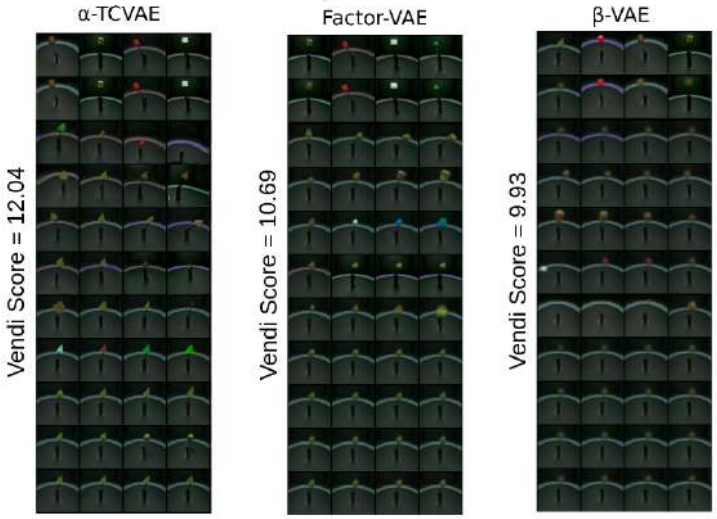


Figure 1.1: Ground truth (first row), reconstructions (second row) and latent traversals comparison of α-TCVAE, Factor-VAE, and β-VAE on the MPI3D-Real Dataset. Notably, α-TCVAE showcases superior visual fidelity and generative diversity, as indicated by a higher Vendi Score.

proposed method consistently outperforms the related baselines, showing a significant improvement in the RL Ant Maze task when applied off-the-shelf in Director, a hierarchical model-based RL agent [111].

1.2. RELATED WORK

Generative Modelling and Disentanglement Recently [96] demonstrated that unsupervised disentangled representation learning is theoretically impossible, nonetheless disentangled VAEs, acting as both representational and generative models, [14, 97, 98, 106] achieve practical results by leveraging implicit biases within the data and learning dynamics [103, 107, 112]. On the generation side, they have been widely used to generate data such as images [113], text [114], speech [115, 116] and music [117]. Various extensions to the base VAE model have been presented to improve generation quality in terms of visual fidelity [118–120]. On the representational side, aiming for explainable and factorized representations, [97] proposed β-VAE, which inspired a number of following disentangled VAE-based models, such as Factor-VAE [98], β-TCVAE [106], and $\beta-$Annealed VAE [121]. Both β-VAE and Factor-VAE aim to learn disentangled representations by imposing a bottleneck on the information flowing through the latent space. While β-VAE does this by introducing a β hyperparameter that increases the strength of the information bottleneck, Factor-VAE introduces a TC regularization term. [106] proposed β-TCVAE, which minimises the total correlation of the latent variables using Monte-Carlo and importance sampling.

[122] proposed the Hausdorff Factorized Support (HFS) criterion, a relaxed disentanglement criterion that encourages only pairwise factorized support, rather than a factorial distribution, by minimizing a Hausdorff distance. This allows for arbitrary distributions of the factors over their support, including correlations between them. Our model, namely α-TCVAE is optimised by a TC lower bound as well, however we do not make use of any trick or expensive sampling strategy. In contrast, we derive a TC lower bound that does not require any extra network or sampling strategy and is theoretically grounded in the Deep Information Bottleneck framework [91].

Disentanglement and Deep Information Bottleneck In the last few years, a link between the latent space capacity and disentanglement of the learned variables [78, 90, 123] has been identified, showing that decreasing the capacity of a network induces disentanglement on the learned representations. This relationship has been explained by the information bottleneck (IB) principle, introduced by [124] as a regularization method to obtain minimal sufficient encoding by constraining the amount of information captured by the latent variables from the observed variable. Variational IB (VIB) [91] has extended the IB framework by applying it to neural networks, which results in a simple yet effective method to learn representations that generalise well and are robust against adversarial attacks. Furthermore, [91, 99] outlined the relationship between VIB, VAE [14] and β-VAE [97], providing an information theoretical interpretation of the Kullback-Leibler (KL) divergence term used in these models as a regularizer. Despite the advantages introduced by the VIB framework, imposing independence between every latent variable may be too strong an assumption [122]. For this reason, [109] introduced the conditional entropy bottleneck (CEB), which assumes conditional independence between the learned latent variables, providing the ability to learn more expressive and robust representations [99]. Recently, a generalization of the mutual Information (MI), namely total correlation (TC), has been used to learn disentangled representations as well [98]. Following [125], who propose a similar TC bound for a multi-view setting, we derive a novel TC lower bound for the unsupervised representational learning setting. As a result, the proposed bound is able to learn expressive and disentangled representations.

Disentanglement and Diversity The ideal generative model learns a distribution that well explains the observed data, which can then be used to draw a diverse set of samples. Diversity is thus an important desirable property of generative models [101]. We desire the ability to produce samples that are different from each other and from the samples we already have at train time, while still coming from the same underlying distribution. The benefits of diversity have been advocated in a number of different contexts, such as image synthesis [126], molecular design [127, 128], natural language text [129], and drug discovery [98]. Motivated by the benefits of generative diversity, several VAE-based models have aimed to show increased diversity in their generated samples [120]. Some works have also noted improvements in diversity due to disentanglement. [130] adversarially disentangle style from content and show enhanced diversity of image-to-image translations. [131] also perform style-content

disentanglement, this time in the context of text generation, and again observe an increase in diversity. [132] shows that disentangling pose, shape, and texture leads to greater diversity in generated images. Collectively, these studies emphasize that diversity is often a valuable indicator of effectiveness in various applications, and suggest that diversity and disentanglement are intertwined aspects of generative models. Yet, to the best of our knowledge, no quantitative analyses that support this claim have been presented. In this work, we present a correlation study, showing how downstream metrics of disentanglement (e.g., DCI [133]) and diversity (e.g., Vendi Score [101]) are correlated across several models and datasets.

1.3. α-TCVAE FRAMEWORK DERIVATION

Motivation. In contrast to most existing methods, which only impose an information bottleneck to learn disentangled representations, we seek to maximise the informativeness of individual latent variables as well. The total joint correlation (TC) can be explicitly expressed in terms of mutual information between the observed data and the latent generative factors, as shown in equation 1.4, allowing us to link disentanglement to latent variables informativeness. As a result, leveraging the TC formulation, we can derive a lower bound that not only promotes disentanglement but also maximises the information retained by individual latent variables.

Derivation. In this section, we formally derive the novel TC bound. Let $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{V}\}$ be the ground-truth set that consists of images $\boldsymbol{x} \in \mathbb{R}^{N \times N}$, and a set of conditionally independent ground-truth data generative factors $\boldsymbol{v} \in \mathbb{R}^M$, where $\log p(\boldsymbol{v}|\boldsymbol{x}) = \sum_k \log p(v_k|\boldsymbol{x})$. The goal is to develop an unsupervised deep generative model that can learn the joint distribution of the data $\boldsymbol{x}$, while uncovering a set of generative latent factors $\boldsymbol{z} \in \mathbb{R}^K$, $K \geq M$, such that $\boldsymbol{z}$ can fully describe the data structure of $\boldsymbol{x}$ and generate data samples that follow the underlying ground-truth generative factors $\boldsymbol{v}$. Since directly optimising the joint TC is intractable, we are going to maximise a lower bound of the joint total correlation $TC(\boldsymbol{z}, \boldsymbol{x})$ between the learned latent representations $\boldsymbol{z}$ and the input data $\boldsymbol{x}$, following the approach proposed by [125]. The total correlation is defined as the KL divergence between the joint distribution and the factored marginals. In our case:

$$TC_\theta(\boldsymbol{z}) \triangleq D_{KL}\left[\int q_\theta(\boldsymbol{z}|\boldsymbol{x}) p_D(\boldsymbol{x}) d\boldsymbol{x} \| \prod_{k=1}^{K} q_\theta(\boldsymbol{z}_k)\right], \tag{1.1}$$

where the joint distribution is $q_\theta(\boldsymbol{z}) = \int q_\theta(\boldsymbol{z}|\boldsymbol{x}) p_D(\boldsymbol{x}) d\boldsymbol{x}$, $p_D(\boldsymbol{x})$ is the data distribution, $q_\theta(\boldsymbol{z}_k) = \int q_\theta(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}_{\neq k}$ and $\boldsymbol{z}_{\neq k}$ indicates that the k-th component of $\boldsymbol{z}$ is not considered. Since we aim to find the encoder $q_\theta(\boldsymbol{z}|\boldsymbol{x})$ that disentangles the learned representations $\boldsymbol{z}$, we can formulate the following objective:

$$TC_\theta(\boldsymbol{z}, \boldsymbol{x}) \triangleq TC_\theta(\boldsymbol{z}) - TC_\theta(\boldsymbol{z}|\boldsymbol{x}), \tag{1.2}$$

where the conditional TC($\boldsymbol{z}|\boldsymbol{x}$) can be expressed as:

$$TC_\theta(\boldsymbol{z}|\boldsymbol{x}) \triangleq \mathbb{E}_{q_\theta(\boldsymbol{z})}\left[D_{KL}\left[q_\theta(\boldsymbol{z}|\boldsymbol{x})\|\prod_{k=1}^{K} q_\theta(\boldsymbol{z}_k|\boldsymbol{x})\right]\right], \tag{1.3}$$

which is the expected KL divergence of the joint conditional from the factored conditionals. Intuitively, we can see that minimizing $TC_\theta(\boldsymbol{z}|\boldsymbol{x})$, $TC_\theta(\boldsymbol{z},\boldsymbol{x})$ will be maximised, enhancing the disentanglement of the learned representation $\boldsymbol{z}$. Moreover, decomposing equation 1.2 we can express the TC in terms of MI [134]:

$$TC_\theta(\boldsymbol{z},\boldsymbol{x}) = \sum_{k=1}^{K} I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) - I_\theta(\boldsymbol{z},\boldsymbol{x}), \tag{1.4}$$

where $I_\theta(\boldsymbol{z},\boldsymbol{x})$ is the mutual information between $\boldsymbol{z}$ and $\boldsymbol{x}$ and is known as the VIB term [91]. Additionally, we can also express it in terms of Conditional MI:

$$TC_\theta(\boldsymbol{z},\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K}\left[(K-1)I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) - I_\theta(\boldsymbol{z}_{\neq k},\boldsymbol{x}|\boldsymbol{z}_k)\right], \tag{1.5}$$

where $I_\theta(\boldsymbol{z}_{\neq k},\boldsymbol{x}|\boldsymbol{z}_k)$ is known as the CEB term [109]. Equation 1.4 and equation 1.5 illustrate the link of the designed objective to both VIB and CEB frameworks. A complete derivation of them can be found in Appendices 1.9 and 1.9, respectively. While the VIB term promotes compression of the latent representation, the CEB term promotes balance between the information contained in each latent dimension. Since we want to promote both disentanglement and individual variable informativeness of the learned latent representation we propose a lower bound that convexly combines the found VIB and CEB terms. We define the bound as follows:

$$TC(\boldsymbol{z},\boldsymbol{x}) \geq \mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\phi(\boldsymbol{x}|\boldsymbol{z})\right] - \underbrace{\frac{K\alpha}{K-\alpha} D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))}_{\text{CEB}} - \frac{(1-\alpha)}{(1-\frac{\alpha}{K})}\underbrace{D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))}_{\text{VIB}}, \tag{1.6}$$

where α is a hyperparameter that trades off VIB and CEB terms. Following [125], we define $r_p(\boldsymbol{z}|\boldsymbol{x}) = N(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p\boldsymbol{I})$ and $r(\boldsymbol{z}) = N(\boldsymbol{0},\boldsymbol{I})$, respectively, where $\boldsymbol{\sigma}_p \triangleq \left(\sum_{k=1}^{K}\frac{1}{\boldsymbol{\sigma}_k^2}\right)^{-1}$ and $\boldsymbol{\mu}_p \triangleq \boldsymbol{\sigma}_p \cdot \sum_{k=1}^{K}\frac{\boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2}$ while $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are the mean and standard deviation used to compute $\boldsymbol{z}_k$ using the reparametrization trick as in [14]. A full derivation of the bound defined in equation 1.6 can be found in Appendix 1.9.

Practical Implications. Disentangled models with M generative factors and K latent dimensions usually have $(K-M)$ noisy latent dimensions [100], but our CEB term induces an inductive bias on the information flowing through every individual latent variable, pushing otherwise noisy dimensions to be informative. The derived TC lower bound generalises the structure of the widely used β-VAE [97] bound. Indeed, for $\alpha = 0$, the TC bound reduces to β-VAE one. A comparison of α-TCVAE,

β-VAE, β-TCVAE, HFS and Factor-VAE lower bounds can be found in Tab. 1.1.

1.4. EXPERIMENTS

In this section, we design empirical experiments to understand the performance of α-TCVAE and its potential limitations by exploring the following questions: (1) Does maximising the informativeness of latent variables consistently lead to an increase in representational power and generative diversity? (2) Do disentangled representations inherently present higher diversity than entangled ones? (3) How are they correlated with other downstream metrics (i.e., FID [135] and unfairness [136])? (4) To what extent does maximising the latent variables' informativeness in disentangled representations improve their downstream usefulness?

Experimental Setup. In order to assess the performance of both proposed and baseline models, we validate the considered models on the following datasets. **Teapots** [137] contains 200,000 images of teapots with features: azimuth and elevation, and object colour. **3DShapes** [112] contains 480,000 images, with features: object shape and colour, floor colour, wall colour, and horizontal orientation. **MPI3D-Real** [110] contains 103,680 images of objects at the end of a robot arm, with features: object colour, size, shape, camera height, azimuth, and robot arm altitude. **Cars3D** [138] contains 16,185 images with features: car-type, elevation, and azimuth. **CelebA** [139] contains over 200,000 images of faces under a broad range of poses, facial expressions, and lighting conditions, totalling 40 different factors. All datasets under consideration consist of RGB images with dimensions 64×64. Among them, CelebA stands out as the most realistic and complex dataset. On the other hand, MPI3D-Real is considered the most realistic among factorized datasets, which we define as those compositionally generated using independent factors. To assess the generated images, we use the FID score [135] to measure the distance between the distributions of generated and real images, and the Vendi score [101] to measure the diversity of sampled images. Both Vendi and FID use the Inception Network [140] to extract image features and compute the related similarity metrics. Since **DCI** [133] scores can produce unreliable results in certain cases, [100, 141, 142], we measure disentanglement using also single neuron classification SNC [141]. Further details on used datasets and metrics are given in Appendix 1.9.

Baseline Methods. We compare α-TCVAE to four other VAE models: β-VAE [97], β-TCVAE [106], β-VAE+HFS [122] and FactorVAE [98], all of which are described in Section 1.2, as well as a vanilla VAE [14]. To assess diversity and visual fidelity beyond VAE-based models, we also compare to a generative adversarial network model, StyleGAN [143].

Generation Faithfulness and Diversity Analyses. We present image generation results from our model alongside baseline models, evaluating performance on the FID and Vendi metrics across datasets. For image generation using VAE-models, we adopt two strategies: (1) Sampling a noise vector from a multivariate standard normal and decoding it. (2) Encoding an actual image, then selecting a latent dimension. The value of this chosen dimension is adjusted by shifts of

+/− 1, 2, 4, 6, 8, or 10 standard deviations. Subsequently, we decode the adjusted representation. In Figures 1.2 and 1.3 the two sampling strategies are labeled as 'Sampled from Noise' and 'Sampled from Traversals' respectively. Figures 1.2 and 1.3 show that α-TCVAE consistently generates more diverse (higher Vendi) and more faithful (lower FID) images than baseline VAE models.

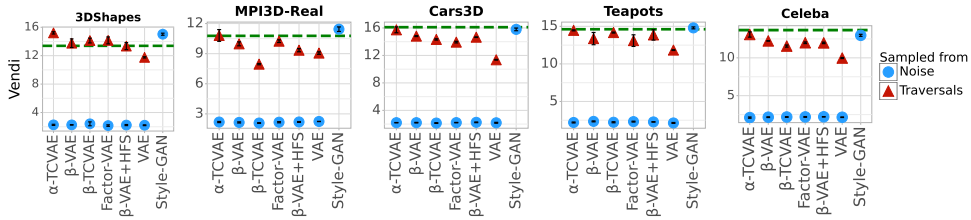


Figure 1.2: Diversity of generated images, as measured by Vendi score. Two different sampling strategies are considered: sampled from white noise and from traversals. The diversity of the images of our model, α-TCVAE, is consistently higher than baseline VAE models, and on par with StyleGAN. The green dashed line represents ground truth dataset diversity. Traversals produce significantly more diverse images than samples.

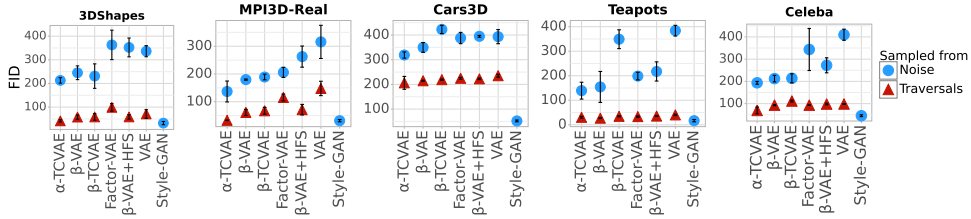


Figure 1.3: Faithfulness of generated images to the data distribution, as measured by FID score. Two different sampling strategies are considered: sampled from white noise and from traversals. The scores for the images of our model, α-TCVAE, are consistently better than baseline VAE models (lower FID is better), and only slightly worse than StyleGAN. Traversals produce significantly more faithful images than samples.

The Vendi score of α-TCVAE is comparable to that of Style-GAN, and its FID score is only slightly worse. Moreover, Style-GAN takes 15x the training time ($\sim$ 2hrs vs. $>$ 30hrs on a single Nvidia Titan XP) and learns only a generative model, whereas VAEs learn both a generative model and a representational model. Noticeably, all VAE-based models perform poorly in terms of both diversity and reconstruction quality when sampling from white noise, highlighting the benefit of a structured sampling strategy when using VAE-based models for generative tasks. Another finding is that traversal-generated images are superior to those obtained from the prior, i.e. sampling from a standard normal and decoding. This is in keeping with

prior work showing that drawing latent samples from a distribution other than the standard normal, e.g. a GMM, often leads to higher quality generated images [144], and it supports the claim that disentangled models allow more systematic exploration of the latent space leading to more diverse images. This claim is also supported by noting that all disentangled VAEs give higher diversity than the vanilla VAE.

Disentanglement Analyses and Downstream Metrics Correlation Study In this section we examine the disentanglement capabilities of α-TCVAE and the related VAE baselines, and how it relates, statistically, with the diversity and quality of generated images, as measured in Section 1.4. Figures 1.4, 1.5 and 1.6 show that α-TCVAE consistently achieves comparable or better DCI, SNC and unfairness scores. The improvement of α-TCVAE over the baselines is most significant on the most realistic factorized dataset, namely MPI3D-Real. Interestingly, while there is a significant gap between the DCI scores of disentangled and entangled models across every factorized dataset, SNC shows that in terms of single neuron factorisation, for both Cars3D and MPI3D-Real, α-TCVAE is the only model that significantly improves over the entangled VAE. This is perhaps due to the tendency of DCI to sometimes overestimate disentanglement [141, 142]. Furthermore, as illustrated in Figure 1.4, no model has been successful in learning disentangled representations from the CelebA dataset. To meaningfully encode CelebA images, we used high-dimensional latent representations (e.g., 48 dimensions). However, as highlighted by [100], disentangling and measuring disentanglement in high-dimensional representations are notoriously challenging tasks. Indeed, while DCI and unfairness present unrealistic results, SNC gave all models a score of zero, and so we do not display the figures here. Figure 1.10 illustrates a significant correlation between the Vendi, unfairness, and DCI metrics. There is a compelling correlation between Vendi and DCI scores, underscoring that diversity and disentanglement are statistically related. This resonates with the understanding that disentangled latent spaces naturally exhibit superior generative diversity [103]. Additionally, Vendi and DCI both exhibit a negative correlation with unfairness. This observation is consistent with [136]'s findings, implying that the fairness of downstream prediction tasks is deeply associated with the diversity and disentanglement of the representations being learned. Further correlations results are given in Appendix 1.9, along with examples of latent traversals.

Attribute Classification Task In this experiment, we train a multilayer perceptron (MLP) to classify sample attributes using the models' encoded latent representations. Figure 1.8 reveals that α-TCVAE either matches or surpasses the baseline models in terms of attribute classification accuracy. The improvement is minor on 3DShapes and Teapots, but more significant on Cars3D and MIP3D-Real. Interestingly, the only dataset where all VAEs exhibit commendable performance is CelebA, where high-dimensional representations are used. This suggests that, for this particular downstream task, the dimensionality of the representation may be the main constraining factor. In fact, this downstream task inherently favours high-dimensional attributes, considering that a MLP is employed for the attribute classification.

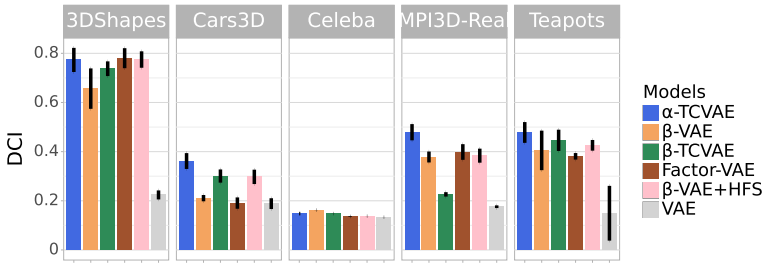


Figure 1.4: Comparison of DCI scores of our model with those of baseline models.

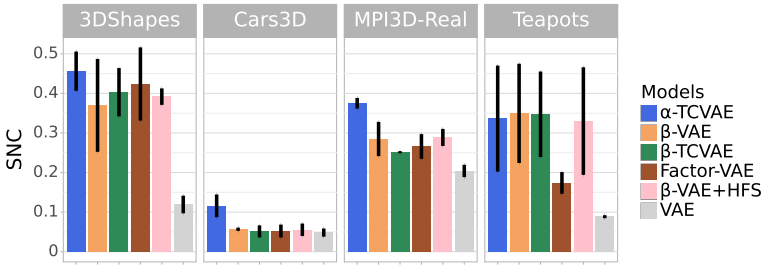


Figure 1.5: Comparison of SNC scores of our model with those of baseline models.

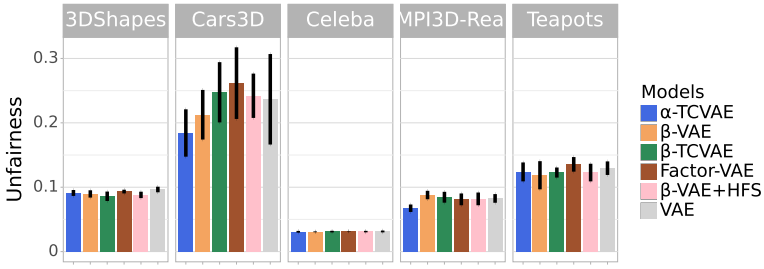


Figure 1.6: Comparison of unfairness scores of our model with those of baseline models.

Loconav Ant Maze Reinforcement Learning Task. In this experiment, a model-based RL agent has to learn its proprioceptive dynamical system while escaping from a maze. Recently, [111] introduced Director, a hierarchical model-based RL agent. Director employs a hierarchical strategy with a Goal VAE that learns and generates sub-goals, simplifying the planning task. The first hierarchy level represents the agent's internal states, while in the second one, the Goal VAE encodes the agent's state and infers sub-goals. As a result, the Goal VAE generates sub-goals to guide the agent through the environment. Given the enhanced generative diversity

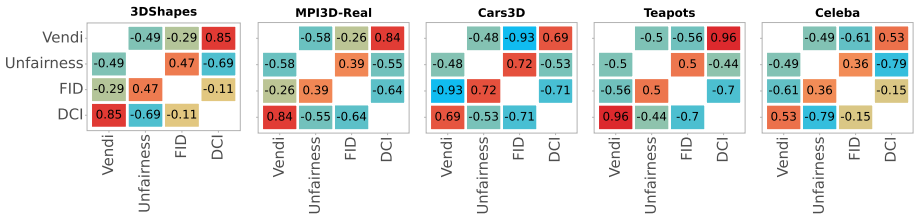


Figure 1.7: Correlations between diversity (Vendi score), generation faithfulness (FID score), unfairness and DCI. Correlations are computed using the results from all models across 5 different seeds.

of α-TCVAE, we postulated that integrating our proposed TC bound could improve Director's exploration. In this experiment, we replaced the beta-VAE objective, used to train Director's Goal VAE, with our TC-bound, expecting a richer diversity in sub-goals, thus expediting environment exploration and enhancing overall learning behaviour. Figure 1.9 compares the performance of Director and Alpha-Director, which replaces β-VAE objective with the proposed TC-bound instead, the results are averaged across three seeds. Figure 1.9-(a) presents the mean return, which scores the performances of the agent on the given task (i.e., finding the exit of the maze while learning proprioceptive dynamics), showing that Alpha-Director significantly outperforms Director, learning faster and to a higher final high mean reward. Figure 1.9-(b) illustrates the Vendi score of sampled goals across batch and sequence length, showing that α-TCVAE generates sub-goals with a higher diversity score. As a result, Alpha-Director has a better exploration, as shown in Figure 1.9-(c), leading to faster learning. Collectively, these findings highlight that α-TCVAE enables the agent to sample a broader range of sub-goals, fostering efficient exploration and ultimately enhancing task performance.

1.5. DISCUSSION AND FUTURE WORK

Through comprehensive quantitative analyses, we answer the defined research questions while delineating the advantages and limitations of the proposed model relative to the evaluated baselines. Our findings resonate with the hypothesis posited by [103], emphasizing a strong correlation between disentanglement and generative diversity. Notably, disentangled representations consistently showcase enhanced visual fidelity and diversity compared to the entangled ones. This correlation persists across all datasets rendered using disentangled representations. Intriguingly, traversal analyses of α-TCVAE, illustrated in Figures 1.1 and 1.16 in Appendix 1.9, reveal that it is able to discover novel generative factors, such as object positioning and vertical perspectives, which are absent from the training dataset. We hypothesize that the CEB term is responsible for this phenomenon. Most existing models optimise only the information bottleneck, and while this can result in factorized representations, it does not directly optimise latent variable informativeness. Our proposed bound also includes a CEB term, and so maximises the average informativeness as well, which

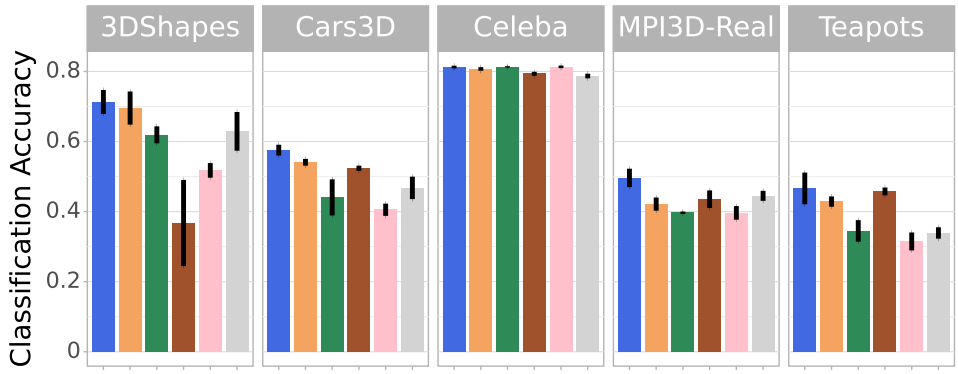


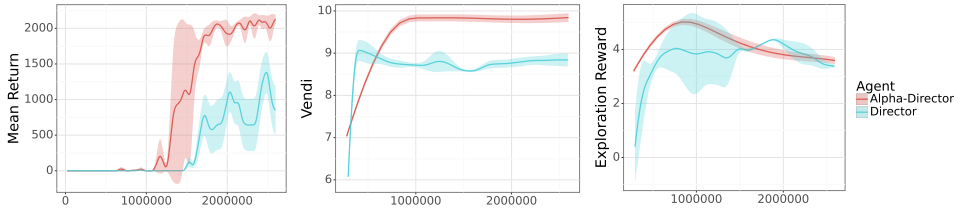
Figure 1.8: Our proposed model either matches or surpasses the baseline models in terms of attribute classification accuracy

may push otherwise noisy variables to learn new generative factors. Future research will delve deeper into comprehending this phenomenon and exploring its potential applications.

In accordance with the literature, the main limitation of α-TCVAE is that, akin to other disentangled VAEs, it is difficult to scale efficiently. This scaling challenge permeates the entire disentanglement paradigm. In high-dimensional spaces, not only do disentangled VAE-based models struggle to produce disentangled representations, but also the metrics used to measure disentanglement tend not to be useful. (e.g., DCI and SNC[133, 141]). On the other hand, disentangled representations have a number of desirable properties, as already showcased in the literature [104]. In particular, their impact is undeniable in the Ant Maze RL experiment from Figure 1.9. Reinforcing this observation, our correlation study underscores the relationship between disentanglement and diversity, leading to the following question: can we leverage diversity as a surrogate for measuring disentanglement in complex and high-dimensional scenarios? We leave the answer to this question as a future work.

1.6. CONCLUSION

We introduce α-TCVAE, a VAE optimised through a convex lower bound on the joint total correlation (TC) between the latent representation and the input data. This proposed bound naturally reduces to a convex combination of the known variational information bottleneck (VIB) [91] and the conditional entropy bottleneck (CEB) [109]. Moreover, it generalises the widely adopted β-VAE bound. By maximizing disentanglement and average informativeness of the latent variables, our approach enhances both representational and generative capabilities. A comprehensive quantitative evaluation indicates that α-TCVAE consistently produces superior representations. This is evident from its performance across key downstream metrics: disentanglement (i.e., DCI and SNC), generative diversity (i.e., Vendi score), visual fidelity (i.e., FID),



(a) (b) (c)

Figure 1.9: Performance of Director, a model-based hierarchical RL agent, and Alpha-Director on the Antmaze task. While director samples sub-goals using the original β-VAE, Alpha-Director samples sub-goals using the proposed α-TCVAE. Sampling using α-TCVAE gives more diverse goals (b), better exploration (c) and significantly higher mean return (a).

and its demonstrated downstream usefulness. In particular, our α-TCVAE showcases significant improvements on the MPI3D-Real dataset, the most realistic factorized dataset in our evaluation, and in a downstream reinforcement learning task. This highlights the strength of maximizing the average informativeness of latent variables, offering a pathway to address the inherent challenges of disentangled VAE-based models.

1.7. Ethic Statement and Reproducibility

To the best of the authors' knowledge, this study does not involve any ethical issues. The authors aim to maximise the reproducibility of the study. The codes of this project will be released in the camera-ready version. In the methods section, notions align with existing literature.

1.8. Acknowledgements

We thank Prof. Yoshua Bengio for the useful feedback provided along the project. We thank Mila - Quebec AI institute, TUDelft, and Compute Canada for providing all the resources to make the project possible.

1.9. APPENDIX

TOTAL CORRELATION LOWER BOUND DERIVATION

In this section we are going to derive the TC lower bound defined in equation 1.6. Since it is defined as a convex combination of marginal log-likelihood, VIB, and CEB terms, we are going to split the derivation into two subsections. First, we will derive a first TC bound that introduces the VIB term. Then, we will derive another TC bound, which explicitly shows the CEB term. Finally, we will define the TC bound shown in equation 1.6 as a convex combination of the two bounds.

TC BOUND AND THE VARIATIONAL INFORMATION BOTTLENECK

Unfortunately, direct optimization of mutual information terms is intractable [91]. Therefore, we first need to find a lower bound of equation 1.4. Following the approach used in [125], we can expand it as:

$$
\begin{aligned}
TC_\theta(\boldsymbol{z}, \boldsymbol{x}) &= \sum_{k=1}^{K} I_\theta(\boldsymbol{z}_k, \boldsymbol{x}) - I_\theta(\boldsymbol{z}, \boldsymbol{x}), \\
&= \sum_{k=1}^{K} \left[\mathbb{E}_{q_\theta(\boldsymbol{x}, \boldsymbol{z}_k)} \left[\log \frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})} \right] \right] - \mathbb{E}_{q_\theta(\boldsymbol{x}, \boldsymbol{z})} \left[\log \frac{q_\theta(\boldsymbol{x}|\boldsymbol{z})}{p_D(\boldsymbol{x})} \right], \\
&= \sum_{k=1}^{K} \left[\mathbb{E}_{q_\theta(\boldsymbol{x}, \boldsymbol{z}_k)} \left[\log \frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})} \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)} \right] \right] - \mathbb{E}_{q_\theta(\boldsymbol{z}, \boldsymbol{x})} \left[\log \frac{q_\theta(\boldsymbol{z}|\boldsymbol{x})}{q_\theta(\boldsymbol{z})} \frac{r(\boldsymbol{z})}{r(\boldsymbol{z})} \right].
\end{aligned}
\tag{1.7}
$$

Let's expand these two terms:

$$
\begin{aligned}
&\mathbb{E}_{q_\theta(\boldsymbol{x}, \boldsymbol{z}_k)} \left[\log \frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})} \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)} \right] = \int\int q_\theta(\boldsymbol{z}_k, \boldsymbol{x}) \log \frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})} \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)} d\boldsymbol{z}_k d\boldsymbol{x}, \\
&= \int\int q_\theta(\boldsymbol{z}_k|\boldsymbol{x}) p_D(\boldsymbol{x}) \left(\log \left(\frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)} \right) + \log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k) - \log p_D(\boldsymbol{x}) \right) d\boldsymbol{z}_k d\boldsymbol{x}, \\
&= H(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z}_k)}[D_{KL}(q_\theta(\boldsymbol{x}|\boldsymbol{z}_k) \| p_\phi(\boldsymbol{x}|\boldsymbol{z}_k))] + \mathbb{E}_{q_\theta(\boldsymbol{z}_k, \boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)].
\end{aligned}
\tag{1.8}
$$

$$
\begin{aligned}
&\mathbb{E}_{q_\theta(\boldsymbol{z}, \boldsymbol{x})} \left[\log \frac{q_\theta(\boldsymbol{z}|\boldsymbol{x})}{q_\theta(\boldsymbol{z})} \frac{r(\boldsymbol{z})}{r(\boldsymbol{z})} \right], \\
&= \int q_\theta(\boldsymbol{x}, \boldsymbol{z}) \log \left(\frac{q_\theta(\boldsymbol{z}|\boldsymbol{x})}{q_\theta(\boldsymbol{z})} \frac{r(\boldsymbol{z})}{r(\boldsymbol{z})} \right) d\boldsymbol{z} d\boldsymbol{x}, \\
&= \int q_\theta(\boldsymbol{z}|\boldsymbol{x}) p_D(\boldsymbol{x}) \left(\left(\log \frac{q_\theta(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})} \right) + \log \left(\frac{r(\boldsymbol{z})}{q_\theta(\boldsymbol{z})} \right) \right) d\boldsymbol{z} d\boldsymbol{x}, \\
&= \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x}) \| r(\boldsymbol{z}))] - \mathbb{E}_{q_\theta(\boldsymbol{x}|\boldsymbol{z})}[D_{KL}(q_\theta(\boldsymbol{z}) \| r(\boldsymbol{z}))].
\end{aligned}
\tag{1.9}
$$

As a result, we can write:

$$TC_\theta(\boldsymbol{z},\boldsymbol{x}) = \sum_{k=1}^{K}\left[H(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z}_k)}[D_{KL}(q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)\|p_\phi(\boldsymbol{x}|\boldsymbol{z}_k))] + \mathbb{E}_{q_\theta(\boldsymbol{z}_k,\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)]\right], \tag{1.10}$$

$$- \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))] + \mathbb{E}_{q_\theta(\boldsymbol{x}|\boldsymbol{z})}[[D_{KL}(q_\theta(\boldsymbol{z})\|r(\boldsymbol{z}))],$$

$$\geq \sum_{k=1}^{K}\left[H(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z}_k,\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)\right] - \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))],$$

$$= \sum_{k=1}^{K}\left[H(\boldsymbol{x}) + \int\left(\int q_\theta(\boldsymbol{z},\boldsymbol{x})d\boldsymbol{z}_{\neq k}\right)\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)d\boldsymbol{z}_k d\boldsymbol{x}\right] - \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))],$$

$$= \sum_{k=1}^{K}\left[H(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z},\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)\right] - \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))],$$

$$= KH(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z},\boldsymbol{x})}[\log\prod_{k=1}^{K} p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)] - \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))],$$

$$= KH(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z},\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}) + \log p_D(\boldsymbol{x})^{K-1}] - \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))],$$

$$= \mathcal{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z})] - \underbrace{\mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))]}_{\text{VIB}} =: \mathcal{L}(\boldsymbol{z},\boldsymbol{x}).$$

Maximizing $\mathcal{L}(\boldsymbol{z},\boldsymbol{x})$ not only maximises the original objective $TC(\boldsymbol{z},\boldsymbol{x})$, but at the same time minimise the gap produced by upper bounding equation 1.10. As a result,

$$\sum_{k=1}^{K}\left[\mathbb{E}_{q_\theta(\boldsymbol{z}_k)}[D_{KL}(q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)\|p_\phi(\boldsymbol{x}|\boldsymbol{z}_k))]\right] + \mathbb{E}_{q_\theta(\boldsymbol{x}|\boldsymbol{z})}[[D_{KL}(q_\theta(\boldsymbol{z})\|r(\boldsymbol{z}))], \tag{1.11}$$

will be minimised, leading to: $r(\boldsymbol{z}) \approx q_\theta(\boldsymbol{z})$ and $p_\phi(\boldsymbol{x}|\boldsymbol{z}_k) \approx q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)$.

Moreover, since $H(\boldsymbol{x})$ and $\log p_D(\boldsymbol{x})^{K-1}$ do not depend on θ, we can drop them from $\mathcal{L}(\boldsymbol{z},\boldsymbol{x})$. Finally, to avoid using a heavy notation, we will denote the VIB term as $D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))$, leading to the first TC bound which introduces the VIB term:

$$TC_\theta(\boldsymbol{z},\boldsymbol{x}) \geq \mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z})] - \underbrace{D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))}_{\text{VIB}}. \tag{1.12}$$

TC bound and the conditional variational information bottleneck

Expanding equation 1.2, we can reformulate $TC(\boldsymbol{z}, \boldsymbol{x})$ as follow:

$$\begin{aligned} TC_\phi(\boldsymbol{z}, \boldsymbol{x}) &= \sum_{k=1}^{K} I_\phi(\boldsymbol{z}_k, \boldsymbol{x}) - I_\phi(\boldsymbol{z}, \boldsymbol{x}), \qquad (1.13) \\ &= \sum_{k=1}^{K} \left[\frac{K-1}{K} I_\phi(\boldsymbol{z}_k, \boldsymbol{x}) + \frac{1}{K} I_\phi(\boldsymbol{z}_k, \boldsymbol{x}) - \frac{1}{K} I_\phi(\boldsymbol{z}, \boldsymbol{x}) \right], \\ &= \sum_{k=1}^{K} \left[\frac{K-1}{K} I_\phi(\boldsymbol{z}_k, \boldsymbol{x}) + \frac{1}{K} \left(I_\phi(\boldsymbol{z}_k, \boldsymbol{x}) - I_\phi(\boldsymbol{z}, \boldsymbol{x}) \right) \right]. \qquad (1.14) \end{aligned}$$

Interestingly, can write the last term of equation 1.14 as:

$$\begin{aligned} I_\phi(\boldsymbol{z}_k, \boldsymbol{x}) - I_\phi(\boldsymbol{z}, \boldsymbol{x}) &= \mathbb{E}_{p_\phi(\boldsymbol{x}, \boldsymbol{z}_k)} \left[\log \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})} \right] - \mathbb{E}_{p_\phi(\boldsymbol{x}, \boldsymbol{z})} \left[\log \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z})}{p_D(\boldsymbol{x})} \right], \qquad (1.15) \\ &= \int \left(\int p_\phi(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}_{\neq \boldsymbol{z}_k} \right) \log \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})} d\boldsymbol{z}_k d\boldsymbol{x}, \\ &\quad - \int p_\phi(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) \log \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z})}{p_D(\boldsymbol{x})} d\boldsymbol{z} d\boldsymbol{x}, \\ &= \int p_\phi(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) \log \frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p(\boldsymbol{x}|\boldsymbol{z})} d\boldsymbol{z} d\boldsymbol{x}, \\ &= - \int p_\phi(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)} d\boldsymbol{z} d\boldsymbol{x}, \\ &= - \int p_\phi(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z}_{\neq k}, \boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)} d\boldsymbol{z} d\boldsymbol{x}, \\ &= -I_\phi(\boldsymbol{z}_{\neq k}, \boldsymbol{x}|\boldsymbol{z}_k). \end{aligned}$$

We can now write equation 1.5:

$$TC_\theta(\boldsymbol{z}, \boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} \left[(K-1) I_\theta(\boldsymbol{z}_k, \boldsymbol{x}) - I_\theta(\boldsymbol{z}_{\neq k}, \boldsymbol{x}|\boldsymbol{z}_k) \right].$$

Interestingly, the second IB term in Eq. (8) can now be expressed as multiple conditional MIs between the observation and $K-1$ other latent variables given the k-th latent representation variable, penalizing the extra information of the observation not inferable from the given latent representation variable. Moreover, we can further

expand the TC as:

$$TC_\theta(\boldsymbol{z},\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K}\left[(K-1)I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) - I_\theta(\boldsymbol{z}_{\neq k},\boldsymbol{x}|\boldsymbol{z}_k)\right], \tag{1.16}$$

$$\begin{aligned}
&= \frac{1}{K}\sum_{k=1}^{K}\left[(K-1)\left[\mathbb{E}_{q_\theta(\boldsymbol{z}_k,\boldsymbol{x})}\left[\log\frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})}\frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}\right]\right]\right.\\
&\left.- \mathbb{E}_{q_\theta(\boldsymbol{x},\boldsymbol{z})}\left[\log\frac{q_\theta(\boldsymbol{z}|\boldsymbol{x})}{q_\theta(\boldsymbol{z}_k|\boldsymbol{x})}\frac{r_p(\boldsymbol{z}|\boldsymbol{x})}{r_p(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathbb{E}_{q_\theta(\boldsymbol{x},\boldsymbol{z})}\left[\log q_\theta(\boldsymbol{z}_{\neq k})\right]\right],\\
&= \frac{1}{K}\sum_{k=1}^{K}\left[(K-1)\left[\mathbb{E}_{q_\theta(\boldsymbol{z}_k,\boldsymbol{x})}\left[\log\frac{q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)}{p_D(\boldsymbol{x})}\frac{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}{p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)}\right]\right]\right.\\
&\left.- \mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))] - \mathbb{E}_{q_\theta(\boldsymbol{x},\boldsymbol{z})}\left[\log\frac{r_p(\boldsymbol{z}_k|\boldsymbol{x})r_p(\boldsymbol{z}_{\neq k}|\boldsymbol{x})}{q_\theta(\boldsymbol{z}_k|\boldsymbol{x})q_\theta(\boldsymbol{z}_{\neq k})}\right]\right],\\
&= \frac{K-1}{K}\sum_{k=1}^{K}\left[H(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z}_k,\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)]\right] - \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))]\right]\\
&+ \frac{K-1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{q_\theta(\boldsymbol{z}_k)}[D_{KL}(q_\theta(\boldsymbol{x}|\boldsymbol{z}_k)\|p_\phi(\boldsymbol{x}|\boldsymbol{z}_k))]\right] \qquad (1.17)\\
&+ \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{q_\theta(\boldsymbol{z}_{\neq k},\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}_k|\boldsymbol{x})\|r_p(\boldsymbol{z}_k|\boldsymbol{x})] + \int D_{KL}(q_\theta(\boldsymbol{z}_{\neq k})\|r_p(\boldsymbol{z}_{\neq k}|\boldsymbol{x}))d\boldsymbol{x},\\
&\geq \frac{K-1}{K}\sum_{k=1}^{K}\left[H(\boldsymbol{x}) + \mathbb{E}_{q_\theta(\boldsymbol{z}_k|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)]\right] - \underbrace{\frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{p_D(\boldsymbol{x})}[D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))]\right]}_{\text{CEB}}.
\end{aligned} \tag{1.18}$$

Maximizing Eq. 1.18 not only maximises the original objective $TC(\boldsymbol{z},\boldsymbol{x})$ but at the same time minimises the gap produced by upper bounding equation 1.17, leading to: $r_p(\boldsymbol{z}_k|\boldsymbol{x}) \approx q_\theta(\boldsymbol{z}_k|\boldsymbol{x})$, $q_\theta(\boldsymbol{z}_{\neq k}) \approx r_p(\boldsymbol{z}_{\neq k}|\boldsymbol{x})$ and $q_\theta(\boldsymbol{x}|\boldsymbol{z}_k) \approx p_\phi(\boldsymbol{x}|\boldsymbol{z}_k)$. Moreover, since $H(\boldsymbol{x})$ does not depend on θ, we can drop it from equation 1.18. Finally, to avoid using a heavy notation, we will denote the CEB term as $D_{KL}(q_\theta(\boldsymbol{z}_k|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))$, leading to the second TC bound which introduces the CEB term:

$$TC_\theta(\boldsymbol{z},\boldsymbol{x}) \geq \frac{K-1}{K}\mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}|\boldsymbol{z})] - \underbrace{D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))}_{\text{CEB}}. \tag{1.19}$$

FINAL TC BOUND

In order to obtain the final expression of the derived TC bound, we can compute a convex combination of the two bounds defined in equation 1.12 and equation 1.19.

$$TC(\boldsymbol{z},\boldsymbol{x}) = (1-\alpha)\left(\sum_{k=1}^{K} I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) - I_\theta(\boldsymbol{z},\boldsymbol{x})\right) \tag{1.20}$$

$$+\alpha\left(\sum_{k=1}^{K}\left[\frac{K-1}{K}I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) + \frac{1}{K}I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) - \frac{1}{K}I_\theta(\boldsymbol{z},\boldsymbol{x})\right]\right), \tag{1.21}$$

$$= \frac{K(1-\alpha)+\alpha(K-1)}{K}\sum_{k=1}^{K} I_\theta(\boldsymbol{z}_k,\boldsymbol{x}) - \frac{\alpha}{K}\sum_{k=1}^{K}\left(I_\theta(\boldsymbol{z},\boldsymbol{x}) - I_\theta(\boldsymbol{z}_k,\boldsymbol{x})\right) - (1-\alpha)I_\theta(\boldsymbol{z},\boldsymbol{x}),$$

$$\geq \frac{K-\alpha}{K}\mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\phi(\boldsymbol{x}|\boldsymbol{z})\right] - \alpha D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x})) - (1-\alpha)D_{KL}(p_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z})),$$

$$= \mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\phi(\boldsymbol{x}|\boldsymbol{z})\right] - \underbrace{\frac{K\alpha}{K-\alpha}D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r_p(\boldsymbol{z}|\boldsymbol{x}))}_{\text{CEB}} - \underbrace{\frac{(1-\alpha)}{(1-\frac{\alpha}{K})}D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z}))}_{\text{VIB}}.$$

where α is a hyperparameter that balances the effects of VIB and CEB terms. Table 1.1 illustrates the lower bounds defined for β-VAE [97], FactorVAE [98], HFS [122] and β-TCVAE [106] comparing them to the derived TC bound. We can see that the three bounds present a similar structure, presenting a marginal log-likelihood term and either one or two KL regularizers that impose some kind of information bottleneck.

ARCHITECTURES AND HYPERPARAMETERS DETAILS

The hyperparameters used for the different experiments are shown in Table 1.2.

All encoder, decoder and discriminator architectures are taken from [122].

FURTHER DETAILS ON DATASETS AND METRICS

DATASETS

We test on five datasets. **Teapots** [137] contains $200,000$ images of size 64×64. Each image features a rendered, camera-centered teapot with 5 uniformly distributed generative factors of variation: azimuth and elevation (sampled between 0 and 2π), along with three RGB colour channels (each sampled between 0 and 1). **3DShapes** [112] consists of $480,000$ images of size 64×64. Every image displays a rendered, camera-centered object with 6 uniformly distributed generative factors of variation: shape (sampled from [cylinder, tube, sphere, cube]), object colour, object hue, floor colour, wall colour, and horizontal orientation, all determined using linearly spaced values. **MPI3D-Real** [110] comprises $103,680$ images of size 64×64. Each image captures objects at a robot arm's end, characterized by 6 factors: object colour, size, shape, camera height, azimuth, and robot arm altitude. **Cars3D** [138] is made up of $16,185$ images of size 64×64. Each image portrays a rendered, camera-centered car, categorized by 3 factors: car-type, elevation, and azimuth. **CelebA** [139]

Model	Lower Bound
β-VAE	$\mathbb{E}_{q_\theta(\boldsymbol{z}\|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}\|\boldsymbol{z})] - \beta D_{KL}(q_\theta(\boldsymbol{z}\|\boldsymbol{x})\|p(\boldsymbol{z}))$
FactorVAE	$\mathbb{E}_{q_\theta(\boldsymbol{z}\|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}\|\boldsymbol{z})] - \beta D_{KL}(q_\theta(\boldsymbol{z}\|\boldsymbol{x})\|p(\boldsymbol{z})) - \gamma D_{KL}(q_\theta(\boldsymbol{z})\|\prod_{k=1}^{K} q_\phi(\boldsymbol{z}_k))$
β-TCVAE	$\mathbb{E}_{q(z\|n)p(n)}[\log p(n\|z) - \alpha I_q(z;n) - \beta D_{KL}(q(z)\|\prod_j q(\boldsymbol{z_j})) - \gamma \sum_j D_{KL}(q(z_j)\|p(z_j))$
HFS	$\mathbb{E}_{q_\theta(\boldsymbol{z}\|\boldsymbol{x})}[\log p_\phi(\boldsymbol{x}\|\boldsymbol{z})] - \gamma[\sum_{i=1}^{K-1}\sum_{j=i+1}^{K} \max_{z\in Z_{:,1}\times Z_{:,2}\times\cdots\times Z_{:,K}} \min_{z'\in Z_{:,(i,j)}} d(z,z')]$
α-TCVAE	$\mathbb{E}_{q_\theta(\boldsymbol{z}\|\boldsymbol{x})}\left[\log p_\phi(\boldsymbol{x}\|\boldsymbol{z})\right] - \frac{K\alpha}{K-\alpha} D_{KL}(q_\theta(\boldsymbol{z}\|\boldsymbol{x})\|r_p(\boldsymbol{z}\|\boldsymbol{x})) - \frac{(1-\alpha)}{(1-\frac{\alpha}{K})} D_{KL}(q_\theta(\boldsymbol{z}\|\boldsymbol{x})\|r(\boldsymbol{z}))$

Table 1.1: This table compares the lower bound objective functions of β-VAE, β-TCVAE, FactorVAE and HFS-VAE. The lower bound objective function of β-VAE is composed of the expected log-likelihood of the data given the latent variables and the KL divergence between the approximate posterior and the prior of the latent variables (i.e., VIB term). The FactorVAE model further adds a KL divergence term between the approximate posterior and the factorized prior of the latent variables, which approximates the total correlation of the latent variables, and HFS-VAE further adds a Monte-Carlo approximation of Hausdorff distance. α-TCVAE, on the other hand, uses a convex combination of VIB term and KL divergence between the approximate posterior and the prior of the latent variables conditioned on the k-th latent variable (i.e., CEB term). K represents the dimensionality of the latent variables, while β, γ and α are hyperparameters of the models.

encompasses over $200,000$ images of size 64×64. Every image presents a celebrity, highlighted by a broad range of poses, facial expressions, and lighting conditions, which sum up to 40 different factors. Every model is trained using a subset containing the 80% of the selected dataset images in a fully unsupervised way. The models are evaluated on the remaining images using the following downstream scores. While CelebA is the most complex dataset, MPI3D-Real is the most realistic among the ones usually used in the disentanglement community.

METRICS

When using the **FID** score to assess image quality, we compare the distribution of generated images to that of the real images. Specifically, FID [135] measures the distance between two distributions of images, and we apply it to measure the distance between the generated images and the real ones. A lower distance is better, indicating that the generated images belong to the distribution of ground truth images.

The **Vendi** score [101], which we use to measure the diversity of the generated images, is computed with respect to a similarity measure. Specifically, it is calculated as the exponential of the entropy of the eigenvalues of the similarity matrix, i.e. the matrix whose (i, j)th entry is the similarity between the ith and jth data points. It can be interpreted as the effective number of distinct elements in the set.

Table 1.2

Dataset	β	γ	α	latent dim K	Training Epochs
Teapots	2	10	0.25	10	50
3DShapes	3	10	0.25	10	50
Cars3D	4	10	0.25	10	50
MPI3D-Real	5	10	0.25	10	50
Celeba	5	10	0.25	48	50

To assess the quality of encoded latent representations, we use DCI, SNC/NK [141] and the unfairness measure of [136].

DCI, the first disentanglement metric we compute, first trains a regressor to predict the generative factors from the latent representation, and from this regressor extracts a matrix of feature importances, where the (i, j)th entry is the import of the ith latent dimension to predicting the jth generative factor. It then takes (a normalized version of) the entropy of rows and columns to compute 'disentanglement' and 'completeness', respectively. The accuracy of the regressor is taken as the 'informativeness' score. The average of these three scores, across all factors and neurons, is the final DCI score.

SNC/NK, the second disentanglement metric we compute, works by first aligning neurons to latent factors using the Kuhn-Munkres algorithm to enforce uniqueness. Then each aligned neuron is used as a classifier for the corresponding factor, by binning its values. A higher accuracy of this single-neuron classifier (SNC) is better, indicating that the factor is well-represented by a single unique neuron. Neuron knockout (NK) is calculated as the difference between an MLP classifier that predicts the generative factor from all neurons, and one that predicts using all neurons but the one that factor was aligned to. A high NK is also better, indicating that no neurons, other than the one it was aligned to, contain information about the given factor. SNC/NK measures a slightly different and stronger notion of disentanglement than DCI, as it explicitly assumes an inductive bias that enforces each factor to be represented by a single latent variable.

MIG, is a disentanglement metric that quantifies the degree of separation between the latent variables and the generative factors in a dataset. It calculates the mutual information between each latent variable and each generative factor, identifying the variable that shares the most information with each factor. The gap, or difference, in mutual information between the top two variables for each generative factor is then computed. A larger gap indicates that one latent variable is significantly more informative about a generative factor than the others, signifying a higher degree of disentanglement. This metric is particularly useful in scenarios where a clear and distinct representation of generative factors is desired in the latent space. MIG thus complements DCI and SNC/NK by providing a measure of how well-separated the representations of different generative factors are within the model's latent space.

EXTENDED RESULTS

Here, we present further results, in addition to those from Section 1.4. Figure 1.10 extended Fig. 1.7, reporting the correlations also with SNC, NK and the attribute classification accuracy as shown in Figure 1.8. Unsurprisingly, there is a strong correlations between the three metrics designed to measure disentanglement: DCI, SNC and NK. This, to some extent, verifies the reliability of these different disentanglement metrics. SNC and NK also correlate strongly with Vendi, as DCI does. This further supports the finding in our paper of a relationship between disentanglement and diversity.

Figure 1.11 shows the results for neuron knockout (NK), the second metric introduced by [141] alongside SNC, which is shown in Figure 1.5. Similar to SNC, the NK score for α-TCVAE is higher than that for baseline VAE models and, while the errorbars often overlap, the superiority of α-TCVAE is consistent across all five datasets and is most substantial on MPI3D-Real. Figure 1.12 shows the results for mutual information gap (MIG), which follows the same trend of NK, SNC and DCI scores. Figures 1.13, 1.14 and 1.15 present the results of Completeness, Disentanglement and Informativeness metrics (DCI-C, DCI-D and DCI-I, respectively). The final DCI scores shown in fig. 1.4 is computed as geomtric mean of the three scores.

DISCOVERING NOVEL FACTOR OF VARIATIONS

Figure 1.16 presents α-TCVAE traversals across 3DShapes, Teapots and MPI3D-Real datasets. The red boxes indicate the discovered novel generative factors, that are not present within the train dataset, namely object position and vertical camera perspective. While we do not have a comprehensive explanation of why such an intriguing phenomenon is shown, we believe that the intuition behind can be explained considering the effects of VIB and CEB terms in the defined bound. Indeed, while VIB pushes individual latent variables to represent different generative factors, CEB pushes them to be informative. As a result, the otherwise noisy dimensions, are pushed to be informative (i.e., CEB) and to represent a distinct generative factor (i.e., VIB), resulting in the discovery of novel generative factors.

RELATIONSHIP BETWEEN CEB AND DIVERSITY

FISHER'S DEFINITION OF CONDITIONAL ENTROPY BOTTLENECK

Fisher's approach to the Conditional Entropy Bottleneck [109] is an extension of the Information Bottleneck (IB) principle [91], aimed at finding an optimally compressed representation of a variable X that remains highly informative about another variable Y, under the influence of a conditioning variable Z. The CEB objective, according to Fisher, is formalized as a trade-off between two competing conditional mutual information terms:

$$\min_{p(z|x)} \left[I(X; Z|C) - \beta I(Y; Z|C) \right]$$

Here, $I(X; Z|C)$ quantifies the amount of information that the representation Z shares with X, conditioned on C. Simultaneously, $I(Y; Z|C)$ measures how much

information Z retains about Y, also under the condition of C. The parameter β serves as a crucial tuning parameter, balancing these two aspects.

ADAPTING CEB TO VAES WITHOUT CONDITIONING VARIABLES

In the realm of Variational Autoencoders, where the training strategy is to reconstruct the input data X using a latent representation Z without any external conditioning C, the CEB framework undergoes a significant simplification. Given that $X = Y$ in a typical VAE setup, the CEB objective reduces to a form where the focus shifts to optimising the mutual information between X and its latent representation Z:

$$\min_{p(z|x)} [(1-\beta)I(X;Z)]$$

This objective can be further broken down as $(1-\beta)(H(X) - H(X|Z))$, where $H(X)$ represents the entropy of the input data, and $H(X|Z)$ is the conditional entropy of the input given its latent representation. This formulation underscores the trade-off between compressing the input data in the latent space and retaining essential information for accurate reconstruction.

INCORPORATING DIVERSITY INTO THE CEB OBJECTIVE

Following [101], Diversity can be quantitatively expressed as the exponential of the entropy of the latent space distribution $q(Z|X)$:

$$\text{Diversity} = \exp(H(Z|X))$$

To understand how the CEB framework relates to this notion of diversity, we utilise the entropy chain rule $\mathrm{H}(Y|X) = \mathrm{H}(X,Y) - \mathrm{H}(X)$, which allows to decompose $H(X|Z)$ in terms of the joint entropy $H(X,Z)$ and the conditional entropy $H(Z)$. Consequently, the CEB objective evolves into a more comprehensive form that explicitly accounts for the diversity of the latent space:

$$\min_{q(z|x)} [(1-\beta)(H(X) - H(X,Z) + H(Z))]$$

$$\min_{q(z|x)} [(1-\beta)(-H(Z|X) + H(Z))]$$

The latter one, makes clear the connection between the CEB term and Diversity as defined in [101]. Indeed, we can see that when minimizing the CEB term the Diversity term is maximised.

DISENTANGLEMENT AND VARIATIONAL INFORMATION BOTTLENECK

Disentanglement in VAEs, following Higgings' β-VAE framework, seeks to learn representations where individual latent variables capture distinct, independent factors of variation in the data. This is achieved by modifying the traditional VAE objective to apply a stronger constraint on the latent space information bottleneck, controlled by a hyperparameter β. The β-VAE, introduced by [97], represents a seminal

approach to disentanglement, promoting the learning of factorized and interpretable latent representations.

On a related front, the Variational Information Bottleneck (VIB) method, formulated by [91], extends the Information Bottleneck principle to deep learning. The VIB approach seeks to find an optimal trade-off between the compression of input data and the preservation of relevant information for prediction tasks. By employing a variational approximation, VIB efficiently learns compressed representations that are predictive of desired outcomes. Interestingly, Alemi formulates a VIB objective that is equivalent to Higgins' β-VAE one. Such result makes evident how imposing a higher information bottleneck leads to higher disentanglement.

SENSITIVITY ANALYSIS OF α

In this section we present a sensitivity analysis of how α affects Vendi and FID results across the considered datasets. To be consistent and analyse how Alpha influences disentanglement scores, we also report a sensitivity analysis of the DCI metric and a correlation study showing how alpha is statistically correlated with FID, Vendi, and DCI metrics.

DIVERSITY AND VISUAL FIDELITY SENSITIVITY ANALYSIS WITH RESPECT TO α

To analyse how α influences the presented results, we performed an evaluation of FID, Vendi and DCI using $\alpha \in [0.00, 0.25, 0.50, 0.75, 1.00]$, where for $\alpha = 0.00$ we obtain β-VAE model, while for $\alpha = 0.25$ we get the results presented in the main paper. Figures 1.17 and 1.18 show that, when $\alpha \in [0.25, 0.50]$ α-TCVAE presents the highest diversity scores, while keeping a FID score comperable to β-VAE.

Interestingly, the two sensitivity analyses show two main trends:

- Diversity increases when using higher values of Alpha.
- FID score improves when using smaller values of Alpha.

Indeed, when using higher values of α , we increase the contribution of the CEB term in equation 1.6, which enhances diversity at the cost of visual fidelity. As a result, the higher the value of α, the more diverse the generated batch of images, and the lower will be the generation quality. However, it can be noticed that when using values of α between 0.25 and 0.50, we get a set of generated images that are more diverse and still have a better or comparable visual fidelity than β-VAE (i.e., α=0).

DISENTANGLEMENT SENSITIVITY ANALYSIS WITH RESPECT TO α

Here, we present a sensitivity analysis of the DCI metric. Figure 1.19 shows that the interval [0.25-0.50] presents higher values of disentanglement, following Diversity and Visual Fidelity analyses that show the best results in the same range. Such a trend can be explained by considering that α weights the contributions of VIB and CEB terms. While the CEB term enhances diversity, the VIB term encourages disentanglement. As a result, we can see that DCI scores decrease when α gets closer to 1. Interestingly, when α is in [0.25,0.50], the combination of CEB and VIB terms

produces a better bound for the Total Correlation objective than when using , which results in higher DCI scores.

Correlation study: how is α correlated with Vendi, FID, and DCI metrics?

Here, we present correlation matrices for all the considered datasets. We computed them using the models trained for the alpha sensitivity analyses. The correlation matrices in fig. 1.20 confirm the trends observed in the other sensitivity analyses (i.e., Vendi, FID, and DCI). Indeed, α has a strong positive correlation with both FID and Vendi, showing that when α increases, diversity increases and FID deteriorates. On the other hand, α has a strong negative correlation with DCI for all datasets besides the Cars 3D dataset, showing that, on average, the higher the value of α, the lower the disentanglement.

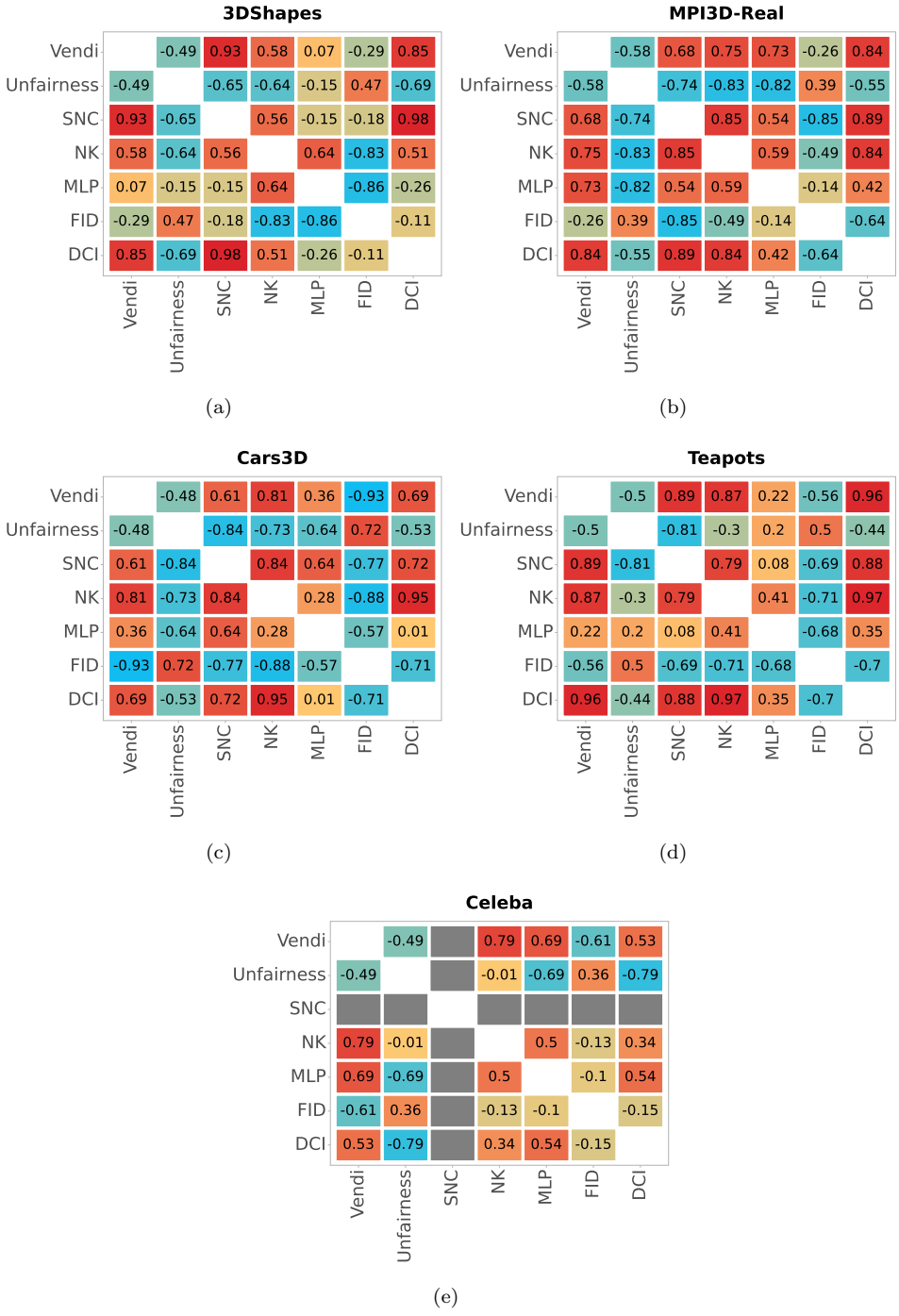


Figure 1.10: Correlations between all metrics we measure, both for the generated images and the representations.

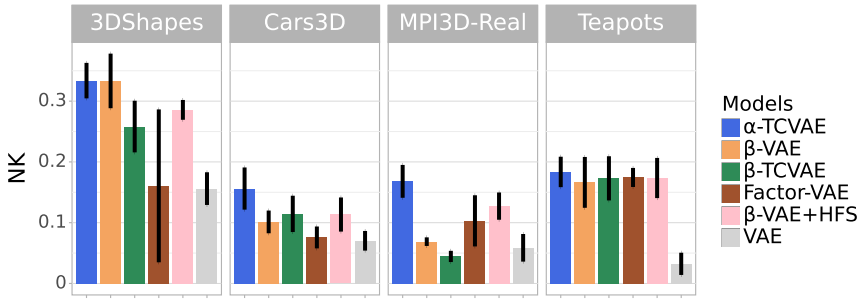


Figure 1.11: Comparison of the neuron-knockout score of α-TCVAE with that of baseline models. As with other metrics presented in the main paper, the improvement of α-TCVAE is minor on 3DShapes and Teapots, but more substantial on Cars3D and MPI3D-Real.

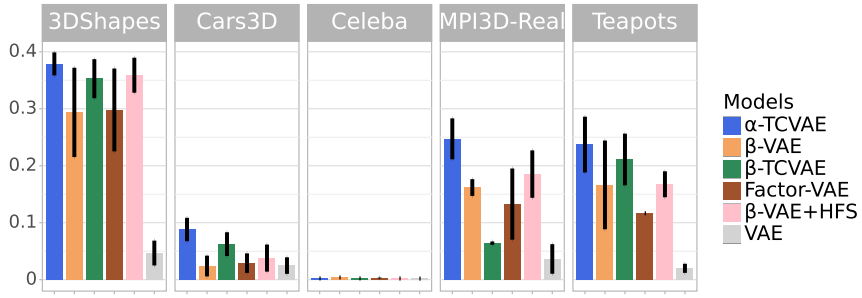


Figure 1.12: Comparison of the MIG score of α-TCVAE with that of baseline models. As with other metrics presented in the main paper, the improvement of α-TCVAE is minor on 3DShapes and Teapots, but more substantial on Cars3D and MPI3D-Real.

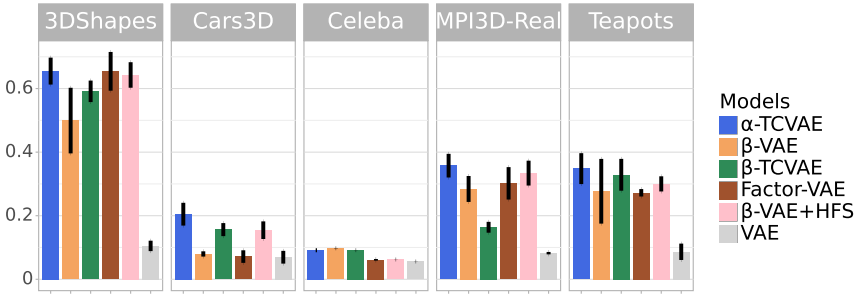


Figure 1.13: Comparison of the DCI-C completeness score of α-TCVAE with that of baseline models. As with other metrics presented in the main paper, the performance of α-TCVAE is comparable on 3DShapes, CelebA and Teapots, and better on Cars3D and MPI3D-Real.

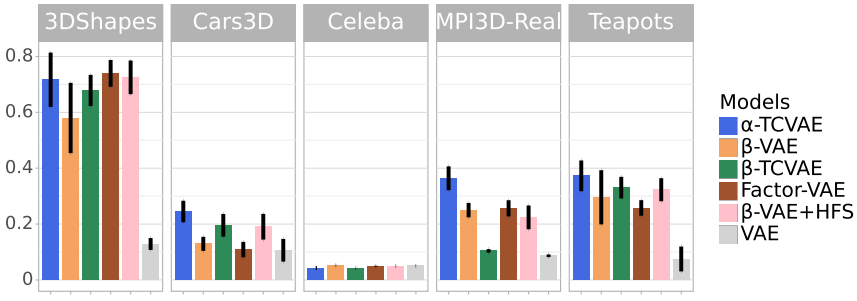


Figure 1.14: Comparison of the DCI-D disentanglement score of α-TCVAE with that of baseline models. As with other metrics presented in the main paper, the performance of α-TCVAE is comparable on 3DShapes, CelebA and Teapots, and better on Cars3D and MPI3D-Real.

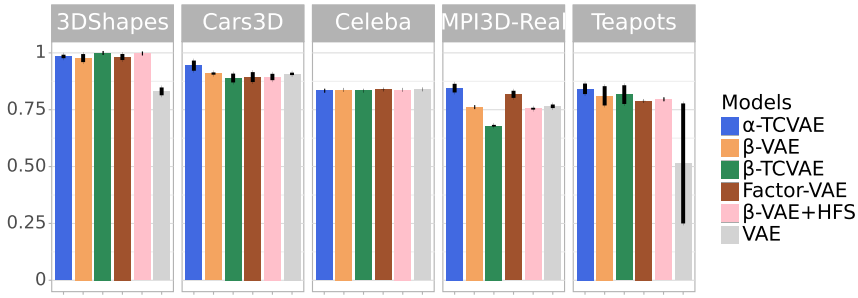
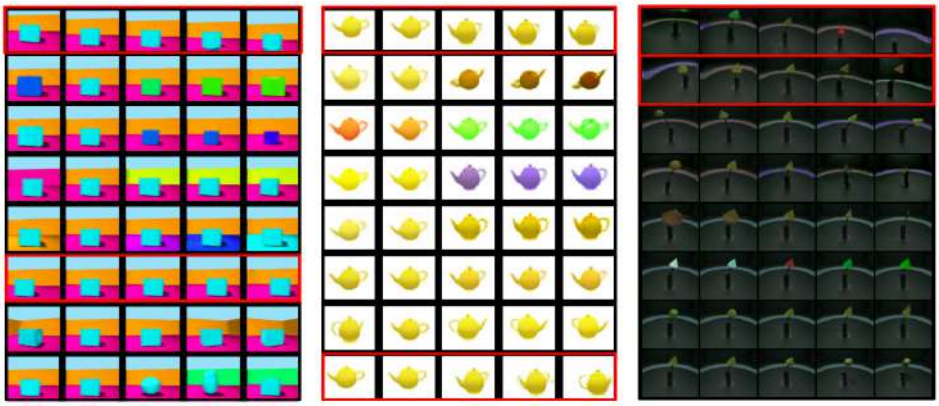


Figure 1.15: Comparison of the DCI-I informativeness score of α-TCVAE with that of baseline models. As with other metrics presented in the main paper, the performance of α-TCVAE is comparable on 3DShapes, CelebA and Teapots, and better on Cars3D and MPI3D-Real.



(a) 3DShapes Traversals (b) Teapots Traversals (c) MPI3D-Real Traversals

Figure 1.16: The generated latent traversals reveal that α-TCVAE can learn and represent generative factors that are not present in the ground-truth dataset, namely vertical perspective and object position. The discovered generative factors are indicated with a red box.

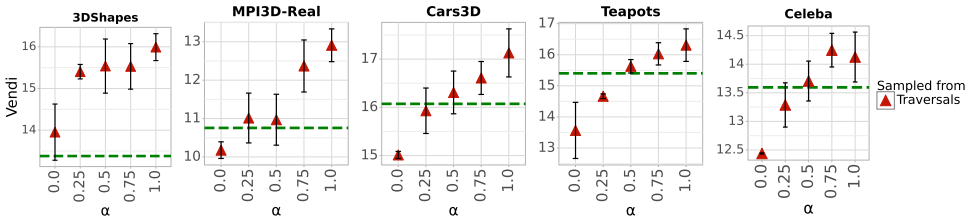


Figure 1.17: Sensitivity Analysis of the Diversity of generated images with respect to α. Only one sampling strategy is considered: sampled from traversals. The green dashed line represents ground truth dataset diversity. It can be seen that the higher alpha the higher will be the Vendi Score.

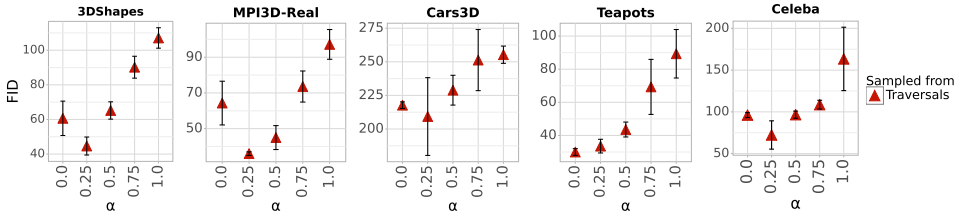


Figure 1.18: Sensitivity Analysis of the Faithfulness of generated images to the data distribution, as measured by FID score, with respect to α. Only one sampling strategy is considered: sampled from traversals. It can be seen that for $\alpha \in [0.25, 0.50]$ the model presents higher visual fidelity.

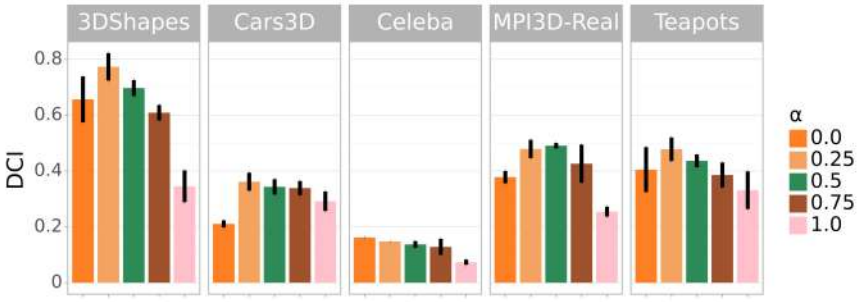


Figure 1.19: DCI scores sensitivity analysis with respect to α. On average when $\alpha \in [0.25, 0.50]$ α-TCVAE presents the best DCI scores.

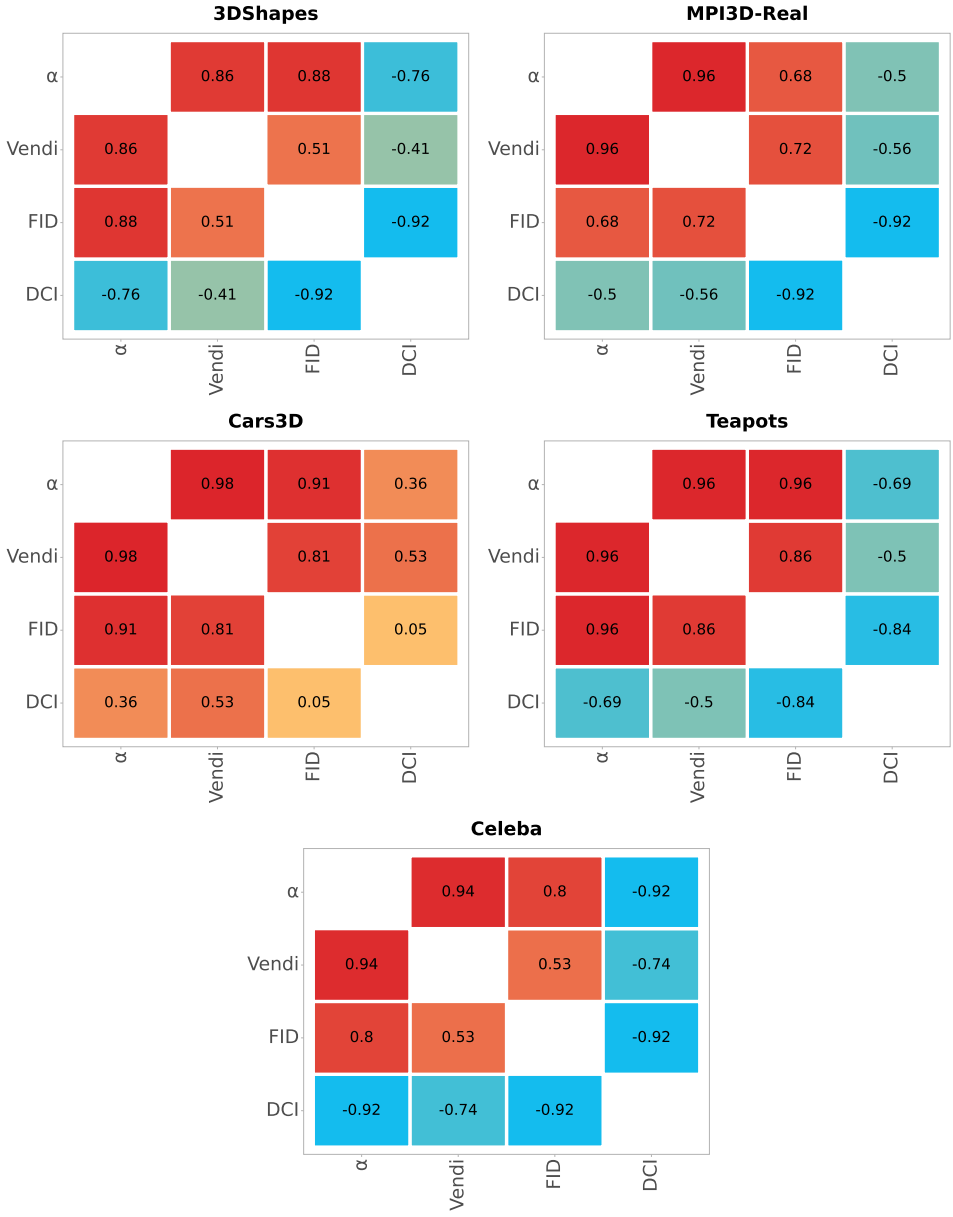


Figure 1.20: Correlations between α, diversity (Vendi score), generation faithfulness (FID score), and disentanglement (DCI). Correlations are computed using the results from all models across 5 different seeds.

2

OBJECT-CENTRIC TEMPORAL CONSISTENCY VIA CONDITIONAL AUTOREGRESSIVE INDUCTIVE BIASES

Published at the NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward

Cristian Meo*
Delft University of Technology, NL
c.meo@tudelft.nl

Akihiro Nakano*
The University of Tokyo, JP

Mircea Lică
Delft University of Technology, NL

Aniket Didolkar
Mila, University of Montreal, CA

Masahiro Suzuki
The University of Tokyo, JP

Anirudh Goyal
Mila, University of Montreal, CA

Mengmi Zhang
Deep NeuroCognition Lab, SG

Justin Dauwels
Delft University of Technology, NL

Yutaka Matsuo
Nanyang Technological University, SG

Yoshua Bengio
CIFAR Chair, Mila, University of Montreal, CA

Unsupervised object-centric learning from videos is a promising approach towards learning compositional representations that can be applied to various downstream tasks, such as prediction and reasoning. Recently, it was shown that pretrained Vision Transformers (ViTs) can be useful to learn object-centric representations on real-world video datasets. However, while these approaches succeed at extracting objects from the scenes, the slot-based representations fail to maintain temporal consistency across consecutive frames in a video, i.e. the mapping of objects to slots changes across the video. To address this, we introduce Conditional Autoregressive Slot Attention (CA-SA), a framework that enhances the temporal consistency of extracted object-centric representations in video-centric vision tasks. Leveraging an autoregressive prior network to condition representations on previous timesteps and a novel consistency loss function, CA-SA predicts future slot representations and imposes consistency across frames. We present qualitative and quantitative results showing that our proposed method outperforms the considered baselines on downstream tasks, such as video prediction and visual question-answering tasks.

2.1. INTRODUCTION

Although recent OC pipelines succeeds at accurately extracting objects from frames in a video [145–147], a persistent problem when applying object-centric models developed for images [66, 148, 149] to videos is temporal consistency. Although learning temporal consistent representations has been a central problem for many years [145, 150–155], learning temporal consistent object-centric representations is particularly difficult as the representations are permutation-equivariant. Prior works utilise various architectural biases to achieve temporal consistency. Some approaches have explored employing prior networks to model temporal consistency expicitly [146, 156–158]. Other models have directly conditioned the slot representations on previous timesteps [155, 156, 159]. However, we argue that architectural biases may not always be enough to achieve temporal consistency. Another approach is to add an auxiliary loss in the representation space [145]. Contrary to [145], we argue that adding such a loss directly on the slots encourages the representations to be too similar between timesteps, which may hinder the model's ability to generalise to longer sequences.

To mitigate the problem of temporal consistency, we propose Conditional Autoregressive Slot Attention (CA-SA), a model-agnostic module that consists of: (1) An autoregressive network that predicts the initial slot representations of the current timestep from the previous timestep, to condition the current slot extraction on prior timesteps, and (2) A temporal consistency loss between the feature-to-slots attention maps of two consecutive frames, to impose the same slot to attend to spatially similar area of the image. Through ablations, we show that the combination of the two is the key to learning a more temporally consistent representations. We present qualitative and quantitative evaluations of the proposed approach on the CLEVRER [160] and Physion [161] datasets, showing how objects' temporal consistency improve in terms

* Equal contributor. NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward.

of downstream task performance.

2.2. Related Works

The problem of temporal inconsistency has been studied for many years [162]. Whenever various image processing algorithms are applied as precursors to video processing, certain temporal inconsistencies can be introduced in the consecutive frames of the video. For example, certain noise reduction algorithms may cause flickering due to slight variations in noise patterns of consecutive frames. To deal with such inconsistencies, previous works have introduced various objectives and priors. [163] introduces a perceptual loss to encourage temporal consistency. [164], introduce two regularization terms that force a frame and its affine transformation to have similar representations. A range of approaches also rely on predicting optical flow or motion information for achieving temporal consistency [165–168]. While these works consider general computer-vision problems, the importance of temporal consistency also applies to video-based object-centric models as well. To ensure temporal consistency various approaches replace the sampling operation, which introduces the permutation equivariance property of slots [66], by conditioning slots on previous ones [155–158]. In this work, besides introducing a novel architectural bias, we introduce an auxiliary loss which enforces consistency by optimising for consecutive attention maps to be similar.

Related works on object discovery and video downstream tasks are summarized in Appendix 2.6.

2.3. Method

When it comes to modelling sequences with an autoregressive model (e.g., RNN [169], autoregressive transformer [5]), ensuring objects-to-slot consistency is necessary to learn meaningful objects dynamics [154, 155, 170]. In contrast to most existing methods, which either enforce an architectural bias [156–158, 171] or a regularizer to enhance temporally consistent object slots [145, 172], our method proposes to use both. Table 2.6 shows a comparison of our proposed method with existing approaches which try to mitigate permutation equivariance property of object-centric representations.

In this section, we present the two main contributions of our proposed approach, namely, (1) CA-SA (Conditional Autoregressive Slot Attention), an autoregressive network that predicts the initial slot representations of the consecutive next timestep and conditions the current slot extraction on prior timesteps, and (2) OPC (Objects Permutation Consistency Loss), an auxiliary loss between two consecutive attention score matrices of the feature-to-slots attention mechanisms, to impose objects permutation temporal consistency between different timesteps. Our proposed objective and architecture are shown in Figure 2.1. As our method is architecture-agnostic, this makes it suitable for any SA-based model for videos.

The overall pipeline, frame generation procedure, and preliminaries about Slot Attention [66] can be found in Appendix 2.6.

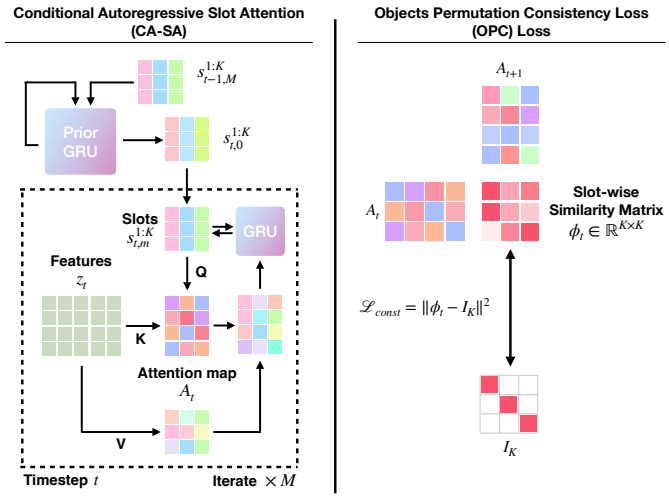


Figure 2.1: Left: Overall CA-SA architecture is represented. The Prior GRU network takes the slots from the previous timestep and condition the initialization of the new slots. The vanilla SA is represented within the dashed box. Right: Visualization of the OPC loss. Two consecutive attention maps A_t, A_{t+1} are used to compute a cosine similarity distance, whose diagonal elements are optimised to match an identity matrix to impose slots' temporal consistency.

2.3.1. CA-SA: Conditional Autoregressive Slot Attention

Conditional Autoregressive Prior. Given an input video consisting of T frames $x_{1:T}$, each input image is first individually encoded via a feature extractor to latent features $z_{1:T}$. Then, features are fed into the Slot Attention architecture to infer slot representations $s_{1:T}^{1:K}$. Since our goal is to model an object-centric dynamics of the environment using slot representations, we need to ensure that the same slots are used to represent the same objects in the scene along the whole video trajectory. In this work, we empirically find that updating individual slots via a Gated Recurrent unit (GRU) based prior network yields the best results:

$$\tilde{s}_t^k, h_t = \text{GRU}_{\text{prior}}(s_{t-1}^k, h_{t-1}) \ , \ k = [1, 2, \cdots, K], \tag{2.1}$$

where $s_t, \tilde{s}_t, h_t$ are the slots, initial slots, and the hidden state of the $\text{GRU}_{\text{prior}}$, respectively. t denotes the timestep. Unlike previous conditioning approaches, which allow for inter-slot interaction using MLP or Transformer, the GRU prior network imposes a structure that prevents representation mixing and preserves the object identity.

OPC: Objects Permutation Consistency Loss. To define a meaningful consistency loss, we draw inspiration from [173]'s findings of how human infants perceive objects using several properties, one of which being their spatiotemporal continuity. For Slot Attention, this principle can be translated into the notion that attention maps generated at consecutive timesteps should exhibit consistency, reflecting the assumption that the same object would persist in spatially proximate pixels across successive frames [174]. However, when defining such consistency within slots, the imposed inductive bias results to be too strong. Indeed, as the loss is backpropagated backward in time through the prior network of slot representations, the cumulative effect of minor alterations in the representations can lead to their deterioration [175].

To solve this issue, we focus on the attention map that is computed within the Slot Attention architecture per timestep. Using attention maps allows us to define a weaker regularization, which does not compromise the slot representation while ensuring slot permutation consistency. Formally, let the attention map at timestep t as $A_t \in \mathbb{R}^{K \times H'W'}$. Given attention maps at consecutive timesteps, A_t, A_{t+1}, to encourage the attention maps for the same slot to be consistent over different timesteps, we define OPC as:

$$\mathcal{L}_{\text{OPC}} = \frac{1}{TK} \sum_{t=1}^{T} \sum_{i=1}^{K} \|(\phi_t - I_K)_{ii}\|^2 \ , \ \phi_t = \frac{A_t A_{t+1}^{\text{T}}}{\|A_t\| \|A_{t+1}\|}, \tag{2.2}$$

where ϕ_t is the attention-wise cosine similarity between consecutive attention maps and $I_K \in \mathbb{R}^{K \times K}$ is an identity matrix. Overall, the proposed method is model-agnostic to any slot-based object-centric learning approach for videos. To incorporate our method, we add the object permutation consistency objective to the original loss function of the method that we wish to apply to. In our case, we optimise the OC feature extractor with a spatial broadcast decoder (SBD) [66, 67] to reconstruct the images from slots. The model is trained using an image reconstruction loss as in [157] together with our proposed objective:

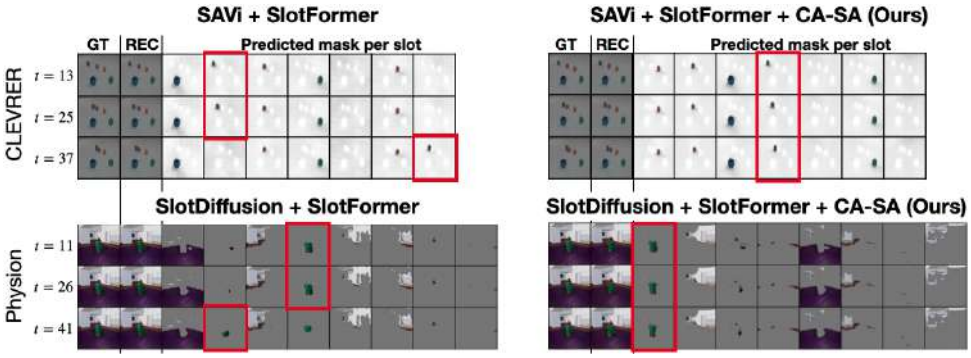
$$\mathcal{L}_{\text{OC-feature extractor}} = \mathcal{L}_{\text{image}} + \lambda \mathcal{L}_{\text{OPC}} \ , \ \mathcal{L}_{\text{image}} = \text{MSE}(x_{1:T}, \hat{x}_{1:T}) \tag{2.3}$$

where λ is a hyperparameter. In our experiments, we set the value to $\lambda = 0.1$ for all datasets.

2.4. EXPERIMENTS

We conduct experiments to evaluate CA-SA by exploring the following question: Do temporally consistent object-centric representations improve their downstream usefulness on video-related tasks? To answer this question, we validate the proposed model on video prediction (VP) and visual question answering (VQA) using CLEVRER [160] and Physion [161] datasets. We also conduct ablational experiments in subsection 2.4.3. We provide further details on the datasets and experimental setup in section 2.6 and section 2.6, respectively.

Figure 2.2: Generation results and predicted masks on CLEVRER (above) and Physion (below). Red square indicate slots which temporal consistency is improved by adding CA-SA.



2.4.1. VIDEO PREDICTION TASK

Table 2.1 and Table 2.2 show the results of the video prediction task for CLEVRER and Physion dataset, respectively. Figure 2.2 shows examples of generated slots for both datasets. On CLEVRER dataset, as the table shows, CA-SA outperforms other baseline models both in terms of visual quality and object-level segmentation. Our model is competitive in terms of visual quality, as the image encoder is the same as in the baseline model. We see that adding temporal consistency improves object-level segmentation for all metrics. On Physion dataset, according to Table 2.2 the proposed model performs slightly worse than SlotDiffusion + SlotFormer for MSE and FVD, while tying for LPIPS with a value of 0.27.

The performance disparity among datasets can be attributed to their distinct characteristics. Most object-centric models are trained with a surplus of slots compared to the total number of objects in the scene [66, 176]. As CLEVRER dataset exhibits a simpler background than Physion, this potentially results in disentanglement with multiple "empty" slots, i.e. slots which attend to neither the foreground objects nor the background [158]. Consequently, models trained on CLEVRER show greater performance enhancements over baseline models due to the possibility of temporal inconsistencies arising from empty slots, whereas achieving temporal consistency is more straightforward on Physion.

Table 2.1: * indicates reproduced results. Best results are indicated in **bold**.

Model	Visual quality			Object dynamics			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	AR % (↑)	ARI % (↑)	FG-ARI %(↑)	FG-mIoU % (↑)
SAVi-dyn	**29.77**	**0.89**	0.19	8.94	8.64	64.32	18.25
SAVi + SlotFormer*	29.22	0.87	0.15	44.19	58.49	65.96	27.90
SAVi + SlotFormer + CA-SA(Ours)	29.47	0.88	**0.14**	**46.50**	**60.52**	**67.25**	**28.60**

Table 2.2: * indicates reproduced results. Best results are indicated in **bold**.

Model	MSE (↓)	LPIPS (↓)	FVD (↓)
STEVE + SlotFormer	832.0	0.43	930.6
SlotDiffusion + SlotFormer*	**489.5**	**0.27**	**737.8**
SlotDiffusion + SlotFormer + CA-SA(Ours)	502.6	**0.27**	759.0

2.4.2. Video Question Answering Task

Table 2.3 and Table 2.4 summarize the results on CLEVRER and Physion VQA tasks, respectively. On CLEVER dataset, adding our proposed method improves the VQA accuracy by 1.9% and 4.6% for accuracy per-option and per-question, respectively. On Physion dataset, our model slightly improves accuracy of both metrics. The detailed results of both datasets are in section 2.6. Again, we observe that the performance gain is larger for CLEVRER dataset, as our model is able to reduce the temporal inconsistency caused by empty slots.

2.4.3. Ablation Study

In this section, we provide ablation results on model architecture of CA-SA. We report visual quality and object dynamics in video object discovery task of CLEVRER dataset in Table 2.5. As the result shows, the combination of using a GRU prior and the proposed auxiliary loss improves over vanilla stochastic SAVi in all metrics except for ARI.

2.5. Conclusion

In this paper, we proposed CA-SA, a model-agnostic module consisting an autoregressive network and OPC, an auxiliary loss aimed to improve object-to-slot temporal consistency of video object-centric models. We experimented on two types of downstream tasks, VP and VQA, and showed that adding our proposed method on top of state-of-the-art baselines improve their performances. Particularly, while we observed a marginal improvement in the video prediction task, CA-SA enhanced the VQA downstream performance across all metrics. We justified such difference considering that, while the VP task relies on the image space, VQA task uses the extracted slots as input space, clearly showing the importance of having temporal consistent slots. As in Slotformer [158], we observed that the two-stage training strategy harms the model's performance at the early rollout steps. Exploring joint training of the base object-centric model and the Transformer dynamics module could potentially benefit the performance of both models. We also leave investigation of combining our method with other video object-centric models and applying our method on wider variation of downstream tasks as future works.

Table 2.3: reporting per-option (per opt.) and per-question (per ques.) accuracy. SF stands for SlotFormer. Both models use Aloe to perform the VQA task. * indicates reproduced results, best ones are in **bold**.

Model	per opt. (%)	per ques. (%)
SF + Aloe*	90.72	80.22
SF + Aloe + CA-SA (Ours)	**92.69**	**84.88**

Table 2.4: reporting accuracy on burn-in (Obs.) and burn-in plus roll-out frames (Dyn.). SD and SF stand for SlotDiffusion and SlotFormer, respectively. * indicates reproduced results, best ones are in **bold**.

Model	Obs. (%)	Dyn. (%)
SD + SF*	63.8	63.9
SD + SF + CA-SA (Ours)	**64.1**	**64.7**

2.6. APPENDIX

OVERALL PIPELINE

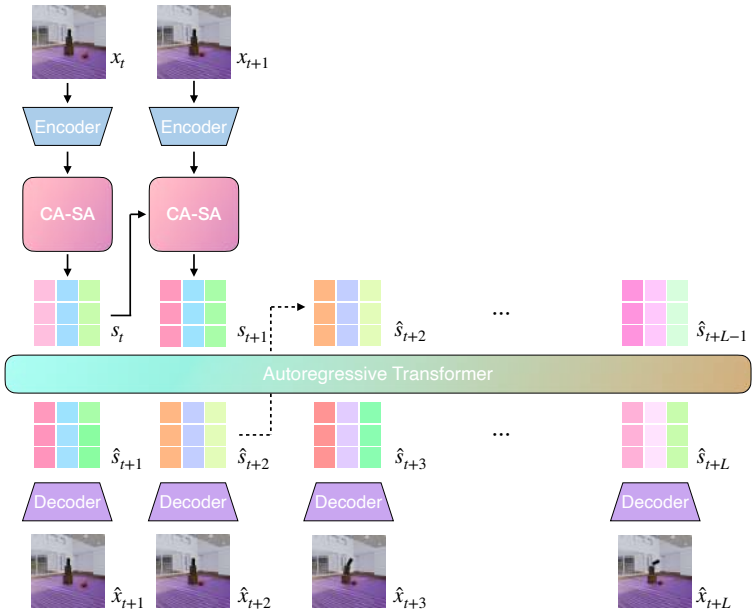


Figure 2.3: Proposed pipeline: Images x_t are first encoded into features, which are used to extract slots s_t. Slots video trajectory is generated using an autoregressive transformer and decoded into the predicted video using a Spatial Broadcast Decoder.

This section presents the overall pipeline, slot attention preliminaries, and frame generation procedure. Figure 2.3 shows the overall pipeline of CA-SA, where CA-SA is used to extract temporal consistent slots from the frames, and the autoregressive

Table 2.5: Ablation study on video object discovery task of CLEVRER dataset.

Model	Prior	Aux. Loss	Visual quality			Object dynamics			
			PSNR (↑)	SSIM (↑)	LPIPS (↓)	AR (↑)	ARI (↑)	FG-ARI (↑)	FG-mIoU (↑)
CA-SA	✔ (GRU)	✔	**40.67**	**0.98**	**0.07**	**78.98**	79.19	**93.94**	**40.71**
CA-SA w/o prior	✗	✔	39.32	0.97	0.08	76.28	79.15	93.83	39.54
CA-SA w/o aux. loss	✔ (GRU)	✗	38.92	0.97	0.08	38.12	61.28	93.60	35.32
StoSAVi	✔ (MLP)	✗	39.81	0.97	0.08	80.47	**79.44**	93.91	40.51

transformer is used to generate future slots.

PRELIMINARIES

Slot Attention (SA). Slot Attention is an architecture proposed for unsupervised object-centric representation learning from images. An input image $x \in \mathbb{R}^{3\times H\times W}$ is processed through a Convolutional Neural Network (CNN) encoder (feature extractor) to extract features $z \in \mathbb{R}^{D_{enc}\times H'\times W'}$. Here, D_{enc} is the feature dimension, and H, W and H', W' are the height and width of the input and encoded image, respectively. The features are then combined with positional embeddings and flattened spatially. Then, the model initializes K, D_{slot}-dimensional object-centric representations, $\tilde{s}^{1:K} \in \mathbb{R}^{K\times D_{slot}}$, from some distribution. Using the slots as query and the features as key and value, Scaled Dot-Product Attention [5] is calculated for M iterations to update slot representations, $s^{1:K} = f_{SA}(\tilde{s}^{1:K}, M)$, where f_{SA} is the SA function. Unsupervised scene decomposition into individual objects is encouraged through the calculation of iterative self-attention by motivating the slots compete against each other to attend to different parts of the image. The slots are then fed to a spatial broadcast decoder (SBD) [67] to reconstruct the input image. The entire architecture is trained using image reconstruction loss only.

FRAME GENERATION USING SLOT REPRESENTATIONS

Given slot representations $s^{1:K}_{1:T}$, we wish to generate a sequence of future slots of length L, $\hat{s}^{1:K}_{T+1:T+L}$. To do so, we follow the approach proposed by SlotFormer [158] and employ an autoregressive transformer $\mathcal{T}$ [5] architecture to perform sequence modelling of the extracted slots. The autoregressive transformer input space is defined using a Multi-Layer Perceptron (MLP) layer, MLP_{in} which maps slots to embeddings, and positional encodings that are summed to the transformer embeddings to impose a temporal structure. A MLP head MLP_{out} is used to map the transformer outputs back to the slot space. Overall, the sequence modelling can be formally expressed as,

$$u^{1:K}_{1:T} = \text{MLP}_{in}(s^{1:K}_{1:T}) \ , \ v^{1:K}_{1:T} = \mathcal{T}(\tilde{u}^{1:K}_{1:T}) \ , \ \hat{s}^{1:K}_{2:T+1} = \text{MLP}_{out}(v^{1:K}_{1:T}), \tag{2.4}$$

where $\tilde{u}^{1:K}_{1:T} = u^{1:K}_{1:T} + p_t$, p_t are the positional encodings [5] and each slot representation is used to predict the same slot at the next timestep. To generate a full trajectory, each slot is generated autoregressively following the approach defined by [158]. The

autoregressive transformer $\mathcal{T}$ is optimised using the following autoregressive objective:

$$\mathcal{L}_{\text{Dyn}} = \text{MSE}(s_{2:T}^{1:K}, \hat{s}_{1:T-1}^{1:K}) + \text{MSE}(x_{2:T}^{1:K}, \hat{x}_{1:T-1}^{1:K}), \tag{2.5}$$

We use the SBD that was trained in paragraph 2.3.1 to decode the predicted slots to image space. While training the autoregressive transformer, the weights of the SBD are kept frozen.

EXTENDED RELATED WORKS

Object-centric learning has been gathering attention as a promising direction towards learning efficient and compositional representations of complex scenes without supervision [66, 148, 172, 177–179]. While recent works have succeeded in applying this approach to real-world scenes [145–147, 149], their evaluation is limited to mask-based metrics. Contrary to this, this work focuses on evaluating the quality of the object-centric representations themselves by applying the representations to downstream tasks. Specifically, we focus on two types of downstream tasks - video prediction and visual question answering. While relatively few, there have been some works that have tackled problems similar to the ones we consider here [146, 155, 158, 180–182]. All of these works, except [146, 158], employ a factored representation coupled with a recurrent dynamics module for video prediction or world modelling. [158] and [146] adopt a transformer-based dynamics module. Out of this, [146, 155, 158] consider Slot Attention [66] as the base model to extract slots from the model. These models rely on architectural priors to impose temporally consistency between slots from neighbouring timesteps. Contrary to these methods, our approach introduces an objective that explicitly optimises for temporal consistency. Moreover, our approach can be integrated into any of the above three approaches. Table 2.6 further highlights the differences between the proposed and existing approaches.

Table 2.6: This table serves to highlight the differences of our models with prior works. The column Temporal-Consistency Prior indicates the approach taken by each model to ensure temporally consistent slot representations across frames in a video.

Method	Temporal-Consistency Prior			Tasks		
	Conditioning of slots	Auxiliary loss	Use RGB inputs only	Reconstruction	Prediction	VQA
SCOFF [159]	previous slots	✗	✓	✓	✓	✓
NPS [155]	previous slots	✗	✓	✓	✓	✗
STEVE [156]	previous slots	✗	✓	✓	✓	✗
SAVi [157]	Transformer prior	✗	✗	✓	✓	✗
VideoSAUR [145]	✗	✓	✓	✓	✗	✗
SlotFormer [158]	MLP/Transformer prior	✗	✓	✓	✓	✓
SlotDiffusion [146]	Transformer prior	✗	✓	✓	✓	✓
Ours	GRU prior	✓	✓	✓	✓	✓

DATASET DETAILS

We validate the proposed model on the Video Prediction and Visual Question Answering downstream tasks on CLEVRER and Physion datasets. In this section, we provide further details on the dataset and preprocessing of the data.

CLEVRER. CLEVRER [160] consists of realistically rendered sequences with multiple 3D objects moving in the scene. The objects differ in shape, color, and texture. The size of each object are kept identical so that no vertical bouncing occurs during collision. The dataset, similar to CLEVR [183] and OBJ3D [184], features smaller objects and more diverse interactions of objects, making it a more challenging task. The attributes of the objects are randomly sampled under the constraint that none of the objects in the scene have the identical attributes. Objects' positions are randomly initialized for each sequence. For each sequence, some objects are randomly chosen such that they cause a collision with each other. The dataset is accompanied by a VQA task with four types of questions: descriptive, explanatory, predictive, and counterfactual. Descriptive questions focus on understanding the video's dynamic content and temporal relations, asking about objects' attributes in an open-ended format. Explanatory questions explore causal relationships, asking which objects or events are responsible for other events. Predictive questions test the ability to predict future events. Counterfactual questions evaluate the understanding of hypothetical scenarios by asking what would or would not happen under altered conditions. Descriptive questions are open-ended questions, while the other three questions are in multiple-choice format with more than one possible answer.

Physion. Physion [161] consists of eight video categories, each showing a different physical phenomenon, such as rigid- and soft-body collisions, falling, rolling, and sliding motions. Each video category presents foreground objects, which vary in categories, textures, colors, and sizes, and diverse background scenes environment showed from randomized camera poses.

The Physion dataset consists of three set: Training, Readout Fitting, and Testing. Following SlotDiffusion [146], we sub-sample the frames by a factor of 3 for training the dynamics module and truncated by 150 frames, since that is the threshold within most of interactions happen. To validate models performances we adopt the official evaluation protocol. First, the dynamics models are trained on videos from the Training set. Then, conditioned by the first 45 frames of Readout Fitting and Test videos, they perform rollout to generate future scene representations. A linear readout model is trained on observed and rollout scene representations from the Readout Fitting set to classify whether an "agent" object (colored in red) contact with the "patient" object (colored in yellow) as the scene unfolds. The classification accuracy of the trained readout model on the Testing set scene representations is reported. For detailed descriptions of the VQA evaluation, refer to their paper [161].

IMPLEMENTATION DETAILS

BASELINES

We build our model on SlotFormer [158] and SlotDiffusion [146] for CLEVRER and Physion dataset, respectively. Their implementations are available online.[1][2]

Stochastic SAVi (StoSAVi). As described by [158], vanilla SAVi [157] occasionally fails to capture objects newly entering the scene. [158] explains that this is caused by the more than one "empty" slots competing against each other to attend to the newly entered object, resulting in multiple slots representing the same object. To solve this problem, [158] proposes a stochastic version of SAVi, in which slots are initialized conditioned on previous timesteps added with a sampling procedure.

Specifically, the output of the prior network is processed through a two-layer MLP with Layer Normalization [185] to predict the mean and log variance of the initial slots at the next timestep:

$$\tilde{s}_t^k \sim \mathcal{N}(\mu_t^k, \{\log \sigma_t^2\}^k) \ , \ (\mu_t^k, \{\log \sigma_t^2\}^k) = \text{MLP}(f_{\text{prior}}(s_{t-1}^k)) \tag{2.6}$$

where f_{prior} is some network used to condition slots on previous timesteps.

The model is optimised by adding a KL divergence loss on the predicted distribution to the image reconstruction loss. The loss only penalizes the log variance with a prior value $\hat{\sigma}$:

$$\mathcal{L}_{\text{KL}} = \frac{1}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} D_{\text{KL}}(\mathcal{N}(\mu_t^k, \{\log \sigma_t^2\}^k) \| \mathcal{N}(\mu_t^k, \{\log \hat{\sigma}^2\}^k)) \tag{2.7}$$

We set $\hat{\sigma} = 0.1$ for all datasets. The coefficient of this loss is set to 1×10^{-4}. We follow the same model architecture as implemented in [158].

SlotDiffusion. The model is trained in two-stage manner, by first pretraining a VQVAE [186] to convert images to tokenized patches, and then train the encoder and Slot Attention architecture. We follow the same model architecture and training settings as [146], where the encoder is a modified ResNet18 encoder [157] and the decoder is LDM-based [19] trained to predict the noise ϵ added to the features z obtained by the pretrained VQVAE.

SlotFormer. After training an arbitrary object-centric model, the slots are extracted for all videos and saved offline. Then, SlotFormer is trained to predict slots at future timesteps, conditioned on burnin frames. The architecture and training strategy are kept unchanged from [158].

[1] https://github.com/pairlab/SlotFormer
[2] https://github.com/Wuziyi616/SlotDiffusion

PROPOSED APPROACH: CA-SA

We implement our prior network using a GRU network. As we implement our method on top of StoSAVi, the initial slots at timestep t are sampled using the predicted mean and log variance which are computed as,

$$\tilde{s}_t^k \sim \mathcal{N}(\mu_t^k, \{\log \sigma_t^2\}^k) \ , \ (\mu_t^k, \{\log \sigma_t^2\}^k) = \text{MLP}(\text{GRU}_{\text{prior}}(s_{t-1}^k)). \tag{2.8}$$

We omit the hidden states h_t for simplicity. Following StoSAVi, the KL divergence loss is added to the total loss.

As described in paragraph 2.3.1, the consistency loss is calculated per slot at each timestep and averaged over them. The coefficient of the loss term is set to $\lambda = 0.1$. To use image reconstruction loss when training the autoregressive transformer and to visualize the predicted slots, we train a CNN-based spatial broadcast decoder separately. This decoder is trained using reconstruction loss in image space, and the loss is not backpropagated to the encoder.

We follow Slotformer [158] approach to evaluate VP and VQA downstream tasks. Specifically, we first train CA-SA, then train the autoregressive Transformer as described in section 2.6 using the inferred slots from the model. We validate both downstream tasks on CLEVRER [160] and Physion [161] datasets.

For CLEVRER dataset, we apply CA-SA on top of SlotFormer [158], while for Physion we use SlotDiffusion [146] as backbone model, as they are the state-of-the-art models on respective datasets. To have a fair comparison we adopt the spatial broadcast decoder used by Slotformer [158] and the conditional latent diffusion model used by SlotDiffusion [146], respectively.

To perform the VQA task, we train an auxiliary model using the slot representations generated by the autoregressive Transformer as inputs. On CLEVRER VQA task, we employ Aloe [187], a Transformer-based architecture that uses slot representations from input frames and text tokens of the question to predict the answer. For predictive questions, we use the trained Transformer to predict slots at future timesteps, and feed them to Aloe. For other questions, we follow the implementation of Aloe. On Physion VQA task, we follow the official protocol by training a readout model on generated slots, as there is no language involved in the task. Following [158], we implement a readout model which consists of a MLP applied on every two slots to extract relations between slots and a max-pool operation which is invariant to input permutations.

On CLEVRER, the training of CA-SA using CNN encoder takes 8 hours to train on 4 V100 GPUs. The training of the autoregressive transformer takes approximately 2 days with the same GPU setup. The training of VQA model takes 3 hours. On Physion, the initial training of VQVAE takes 20 hours. The training of SlotDiffusion requires 30 hours of training on 8 A100 GPUs. The training of the autoregressive transformer takes approximately 15 hours on 4 V100s. The training of the readout model finishes in less than 5 minutes.

Table 2.7 and Table 2.8 describes the hyperparameters used in our the experiments.

Table 2.7: Hyperparameters used to train different encoders on each dataset.

Dataset	CLEVRER	Physion
Image encoder	ResNet18	ResNet18
Image resolution (H, W)	$(64, 64)$	$(128, 128)$
Length of sequence T	6	3
# of features $H'W'$	4096	1024
Feature dimension D_{enc}	128	192
# of slots K	7	8
# of slot attention iteration M	3	2
Slot dimension D_{slot}	128	192
Batch size	64	48
Training epochs	12	10

EXPERIMENTAL SETUP

Video Prediction Task. We compare CA-SA with three state-of-the-art, OC models, SAVi-dyn, SAVi + SlotFormer, and SlotDiffusion + SlotFormer. SAVi-dyn uses SAVi [157] as the encoder and combines with a Transformer-LSTM to generate future slots. SAVi + SlotFormer and SlotDiffusion + SlotFormer combine respective models. For CLEVRER, the stochastic verison of SAVi was used in order to accomodate to new objects entering the scene during rollout.

For CLEVRER, we use PSNR, SSIM, and LPIPS to evaluate the visual quality of the frames generated by each model, and ARI, FG-ARI, FG-mIoU, and AR for evaluation of object-level segmentation quality. For Physion, following [146], we report visual quality metrics only, MSE, LPIPS, and FVD [188].

We follow [146, 158] with the evaluation protocol for both datasets. On CLEVRER, we use 6 burn-in timesteps to condition the model and then perform a rollout to predict the next slots for 10 steps. On Physion, the model was trained using 15 burn-in and 10 rollout timesteps. The predicted slots were decoded to images using the SBD and compared with the ground truth ones.

Video Question Answering Task. For both datasets, we apply CA-SA on top of their respective state-of-the-art model. On CLEVRER, we compare against SlotFormer + Aloe (denoted as SF + Aloe) [158]. SF + Aloe first trains StoSAVi as the feature extractor, followed by SlotFormer. Then, the predicted slots from SlotFormer and text tokens of the question are used to train Aloe, a Transformer-based VQA model. For Physion, we select SlotDiffusion + SlotFormer as the baseline model (SD + SF) [146]. This model first trains SlotDiffusion as the feature extractor, followed by SlotFormer, and finally a readout model using the predicted slots.

We report two types of average accuracy on CLEVRER VQA task, per-option (per opt.) and per-question (per ques.), as the VQA task includes multiple choice questions with more than one possible answers. The per option accuracy assesses the model's overall correctness in selecting individual options across all questions.

Table 2.8: Hyperparameters used to train autoregressive transformer on each dataset.

Dataset	CLEVRER	Physion
Burnin frames T	6	10
Rollout frames L	15	10
Batch size	64	128
Training epochs	80	25
# of layers	4	12
# of heads	8	8
Dimension	256	256
FFN dimension	1024	1024

Conversely, the per question accuracy measures correctness on a question-by-question basis, necessitating the accurate selection of all answer choices for each question. For Physion VQA task, we report the accuracy when using only burn-in frames (denoted as Obs.) and using burn-in frames and rollout frames (Dyn.).

We follow the implementation of [158] for evaluation on both datasets. On CLEVRER, we train Aloe [187] using the predicted slots by SlotFormer, generated by the procedure described in section 2.6. The slots are concatenated with the text tokens of the questions and then fed to Aloe. On Physion, we train a readout model which receives every two predicted slots at each timestep as inputs. The outputs of the readout model are max-pooled over all pairs of slots and time to predict the answer.

ROLLOUT VISUALIZATIONS

We provide further qualitative results of generated results and predicted attention maps on CLEVRER and Physion datasets in Figure 2.4 and Figure 2.5, respectively.

ADDITIONAL RESULTS ON VQA TASK

We provide the accuracy per type of questions of CLEVRER in Table 2.9. We report the per-scenario accuracy on Physion for the model trained with rollouts in Table 2.10.

Table 2.9

Model	Descriptive	Explanatory		Predictive		Counterfactual	
		per opt. (%)	per ques. (%)	per opt. (%)	per ques. (%)	per opt. (%)	per ques. (%)
Aloe + SlotFormer*	93.67	95.10	86.44	93.26	83.25	83.79	57.52
Aloe + SlotFormer + CA-SA(Ours)	**94.10**	**96.56**	**90.65**	**94.85**	**90.28**	**86.65**	**64.47**

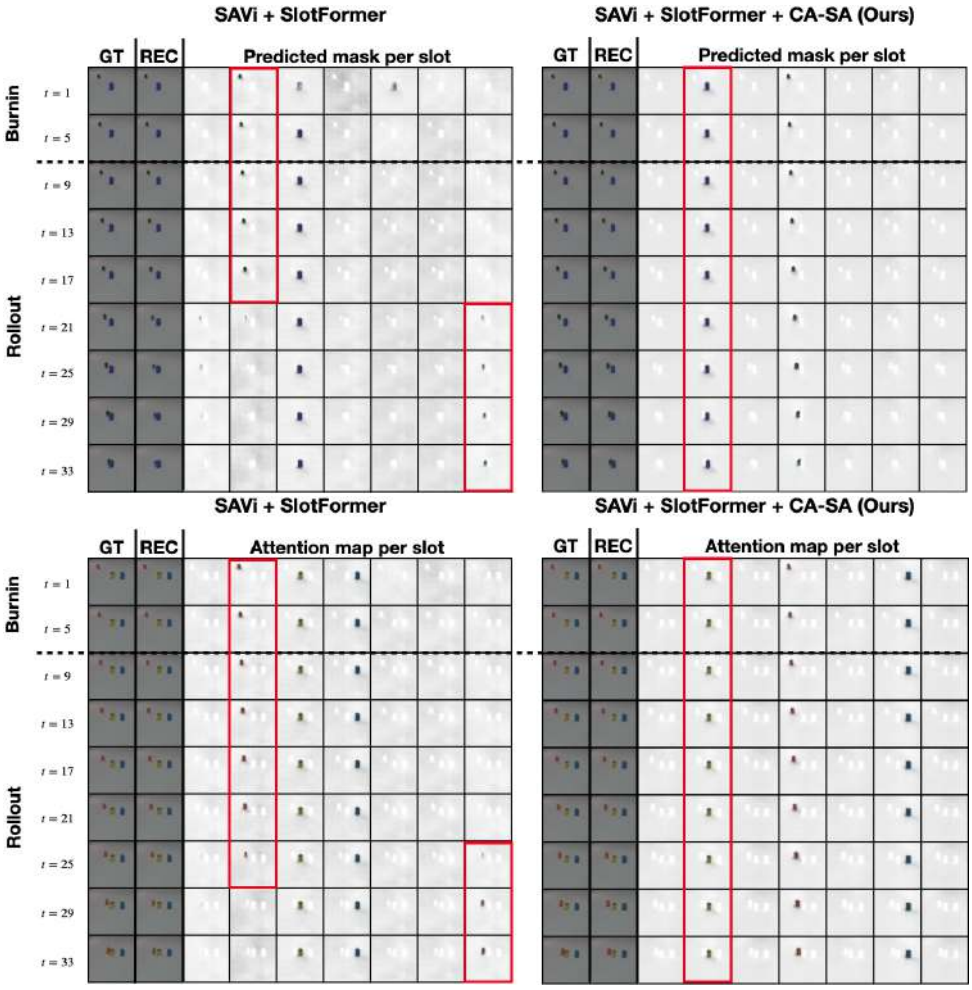


Figure 2.4: More generation results and predicted masks on CLEVRER. Red square indicate slots which temporal consistency is improved by adding CA-SA.

Table 2.10

Model	Collide	Contain	Dominoes	Drape	Drop	Link	Roll	Support	Avg.
SlotDiffusion + SlotFormer*	**75.3**	63.3	49.2	51.3	**65.3**	59.3	**68.0**	70.0	63.9
SlotDiffusion + SlotFormer + CA-SA(Ours)	68.7	**64.0**	**51.6**	**66.0**	60.0	**64.7**	62.7	**72.7**	**64.7**

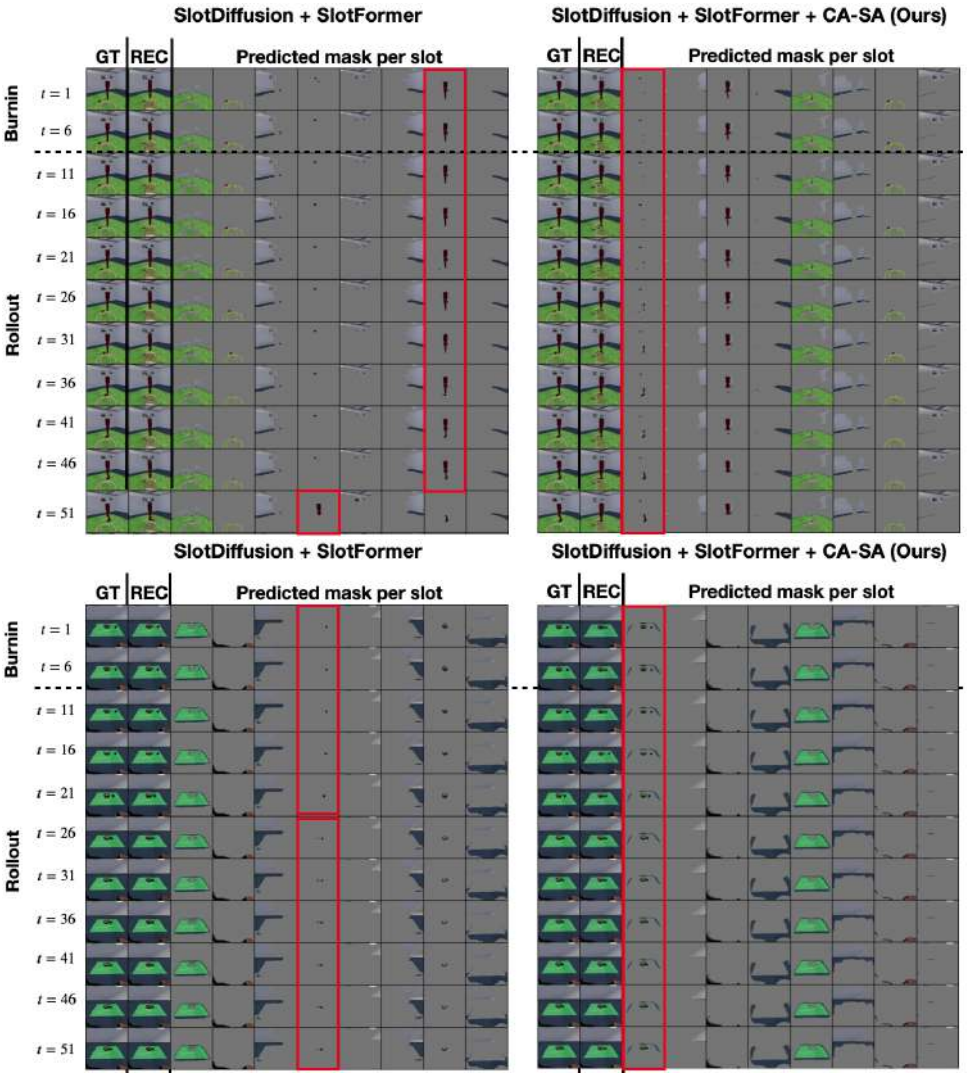


Figure 2.5: More generation results and predicted masks on Physion. Red square indicate slots which temporal consistency is improved by adding CA-SA.

3

EXTREME PRECIPITATION NOWCASTING USING TRANSFORMER-BASED GENERATIVE MODELS

Published as **Spotlight publication** at the ICLR 2024 Workshop on Tackling Climate Change with Machine Learning

Cristian Meo* *Delft University of Technology, NL* c.meo@tudelft.nl

Ankush Roy* *Delft University of Technology, NL*

Mircea Lică* *Delft University of Technology, NL*

Junzhe Yin *Delft University of Technology, NL*

Zeineb Bou Cher *Delft University of Technology, NL*

Yanbo Wang *Delft University of Technology, NL*

Ruben Imhoff *Deltares, NL*

Remko Uijlenhoet *Deltares, NL*

Justin Dauwels *Delft University of Technology, NL*

This paper presents an innovative approach to extreme precipitation nowcasting by employing Transformer-based generative models, namely NowcastingGPT with Extreme Value Loss (EVL) regularization. Leveraging a comprehensive dataset from the Royal Netherlands Meteorological Institute (KNMI), our study focuses on predicting short-term precipitation with high accuracy. We introduce a novel method for computing EVL without assuming fixed extreme representations, addressing the limitations of current models in capturing extreme weather events. We present both qualitative and quantitative analyses, demonstrating the superior performance of the proposed NowcastingGPT-EVL in generating accurate precipitation forecasts, especially when dealing with extreme precipitation events. The code is available at https://github.com/Cmeo97/NowcastingGPT.

3.1. INTRODUCTION

The advent of climate change has escalated the frequency of intense rainfall events across various regions worldwide, leading to considerable societal and infrastructural impacts [189–193]. Consequently, the ability to accurately forecast short-term shifts in rainfall patterns is gaining importance, attracting a growing body of research focus [194–197]. The field of precipitation nowcasting, which involves predicting rainfall changes within a six-hour window, plays a crucial role in enabling timely responses to these rapid meteorological variations [193, 198–200]. In the context of escalating climate change impacts, the field of precipitation nowcasting is increasingly vital for mitigating the adverse effects of intense rainfall events. This research area empowers the development of advanced forecasting models that can provide accurate, short-term rainfall predictions. Such capabilities are essential for proactive disaster management and climate resilience strategies, enabling communities and infrastructure planners to prepare for and respond to extreme weather events more effectively, thereby contributing to meaningful efforts in addressing the climate crisis.

3.2. RELATED WORKS

Conventional nowcasting techniques, exemplified by frameworks such as PySTEPS [201], adopt the ensemble-based methodology reminiscent of Numerical Weather Prediction (NWP) to incorporate uncertainty while modeling precipitation dynamics through the lens of the advection equation [202]. On the other hand, Deep learning-based approaches, leveraging extensive datasets of radar observations, can be trained without the constraints of predefined physical assumptions, significantly enhancing forecast accuracy [202]. In the last few years, precipitation nowcasting using deep learning models has been cast as a video prediction problem [196, 197, 203, 204], where given an input spatio-temporal sequence of N frames $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{N\times H\times W\times C}$, H, W denote the spatial resolution and C represents the image channels or the different type of measurements (e.g., radar, heat maps, etc), the goal is to predict the next M frames $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{M\times H\times W\times C}$. Among the most notable advancements in the field, Generative Adversarial Networks [GAN; 15] have emerged as a powerful approach, exemplified by methods such as DGMR [202], which employs both spatial

and temporal discriminators to ensure the fidelity of generated sequences to the ground truth. Moreover, Transformer-based strategies [5] leverage an Autoregressive Transformer (AT) to model the hidden dynamics of precipitation maps [203, 205]. For instance, [203] employs Nuwä [206], an AT that uses a sparse attention mechanism, namely 3DNA [206], to adeptly capture the complexities of precipitation dynamics. Moreover, [203] regularizes the hidden dynamics incorporating an Extreme Values Loss (EVL) to effectively model and predict extreme precipitation events, which are notoriously difficult to represent and predict. Although these models have improved in terms of prediction capabilities, they present critical drawbacks. Firstly, the prediction quality degrades very quickly, resulting in predicted sequences that are inconsistent over time. Secondly, the time required to generate the predicted sequences is extremely high, which is a critical problem considering that nowcasting predictions are supposed to predict the very next future. For instance, Nuwä-EVL [203] takes over 5 minutes to predict the next precipitation maps on a Nvidia RTX A6000. Furthermore, predicting and representing extreme precipitation events is still very challenging for all the proposed models. Although [203] uses an EVL as a regularizer, it assumes a predefined set of representations that should embed the extreme events features, assuming that the extreme features never change during training. Since the topology of the hidden space changes during training, we believe that using fixed representations is a wrong inductive bias. In this work, we propose NowcastingGPT, which follows VideoGPT framework [207], employing a Vector Quantized-Variational AutoEncoder (VQ-VAE) [186] to extract discrete tokens and an Autoregressive Transformer [208] to model the hidden dynamics. Moreover, we propose a novel approach to correctly compute the EVL regularization without assuming any fixed extreme representation. Moreover, we benchmark TECO [209], an efficient transformer-based video prediction model that generates temporally consistent frames, on the precipitation nowcasting task. Finally, we present both qualitative and quantitative comparisons of the considered models.

3.3. METHODOLOGY

Video prediction tasks, at their core, involve forecasting the future frames of a video sequence based on past observations, akin to predicting the next scenes in a dynamic storyline. This challenge extends naturally to nowcasting, where the goal is predicting satellite imagery or radar maps, capturing the evolution of environmental and weather conditions over time. Both domains share the fundamental task of modeling and anticipating the progression of complex, time-varying patterns, making techniques developed for video prediction highly relevant and applicable to the realm of nowcasting.

3.3.1. NOWCASTING AS VIDEO PREDICTION

Video prediction tasks are known for their sample inefficiency, which poses significant challenges in learning accurate and reliable models. To address this, recent advancements have introduced spatio-temporal state space models, which typically consist of a feature extraction component coupled with a dynamics prediction module. These

models aim to understand and predict the evolution of video frames by capturing both spatial and temporal relationships. Notable examples include Nuwä [206] and VideoGPT [207] which, leveraging the space-efficient VQ-VAE feature extraction, and the powerful sequence modeling capabilities of Autoregressive Transformers, can achieve a deeper understanding of the underlying video dynamics, leading to more accurate predictions of future frames. We define the video prediction backbone of the proposed nowcasting model following the VideoGPT framework, using a VQ-VAE as a feature extractor and an Autoregressive Transformer [208] to learn the latent space dynamics and predict the future precipitation maps. A detailed description of the NowcastingGPT model, depicted in Fig. 3.1, can be found in appendix 3.6.

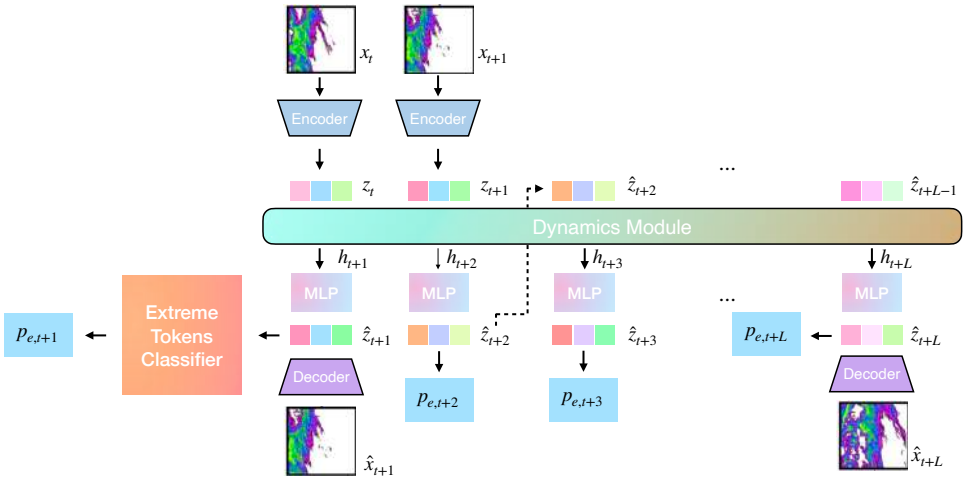


Figure 3.1: The VQ-VAE Encoder and Decoder are depicted in blue and purple respectively. The Extreme tokens classifier is depicted in orange, it takes the predicted tokens as input from the transformer and outputs the probabilities $P_{e,t}$ used in the EVL loss.

3.3.2. Extreme Value Loss Regularization

When dealing with imbalanced data, the standard cross-entropy loss often falls short, particularly when classifying extreme events. To address this, the Extreme Value Loss (EVL) has been introduced as a more effective alternative, designed to balance the disparities between extreme and non-extreme cases in time series data [210]:

$$\mathrm{EVL}(u_t, v_t) = -\beta_1 \left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log(u_t) - \beta_0 \left[1 - \frac{1-u_t}{\gamma}\right]^{\gamma} (1-v_t)\log(1-u_t), \quad (3.1)$$

where v_t represents the ground truth labels (extreme/not extreme), u_t the predicted probabilities, and γ, a hyperparameter of the generalised Extreme Value (GEV) distribution. By incorporating β_0 and β_1, which reflect the proportions of non-extreme and extreme events, EVL effectively balances the learning process. When regularizing an Autoregressive Transformer the EVL enhances the model's ability to predict and represent extreme events. To this end, we define a classifier that dynamically predicts extreme labels. As a result, we can use the EVL to regularize the Autoregressive Transformer learning behaviour and improve its ability to capture extreme phenomena in data sequences. A detailed description of the classifier can be found in appendix 3.6, while the full derivation of the EVL loss can be found in appendix 3.6.

3.4. EXPERIMENTS

In this section, we design empirical experiments to understand the performance of NowcastingGPT-EVL and its potential limitations by exploring the following questions: (1) Does EVL regularization improve the nowcasting performances of the proposed model? (2) How does time consistency affect downstream results? (3) Does learning extreme representations provide a more effective inductive bias compared to relying on predefined ones?

3.4.1. DATASET AND EXPERIMENTAL SETUP

Our nowcasting study aims to predict precipitation patterns up to three hours into the future. This approach generates a series of six future precipitation maps, each separated by 30 minutes, conditioned on three previous precipitation maps used as input. Specifically, we use radar maps defined 256×256 images, which follow the approach used in [203]. More details about the dataset are described in appendix 3.6. We compare the proposed model to a classic benchmark, namely Pysteps [201], a temporally consistent video prediction benchmark, TECO [209], the NowcastingGPT model which is described in the appendix 3.6 and Nuwä-EVL proposed by [203], which uses fixed latents to represents extreme features. An in-depth description of the considered baselines can be found in appendix 3.6. To quantitatively assess the experiments we use visual fidelity metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and Pearson Correlation Score (PCC), and nowcasting metrics, such as Critical Success Index (CSI), False Alarm Rate (FAR) and Fractional Skill Score (FSS). Since fidelity metrics cannot capture extreme event classification, we plot an ROC curve of the extremes to assess the considered baselines in terms of extreme classification capabilities. A detailed description of these metrics can be found in appendix 3.6. Table 3.1 presents a quantitative comparison between all proposed methods in terms of number of parameters, training and generation efficiency. Interestingly, TECO, showcases orders of magnitude more efficient generation time and cuts the training time by approximately 100 hours compared to its closest counterpart. Furthermore, with a generation time of 322.86 seconds, Nuwä-EVL constitutes a good indicator for the sampling efficiency of autoregressive models.

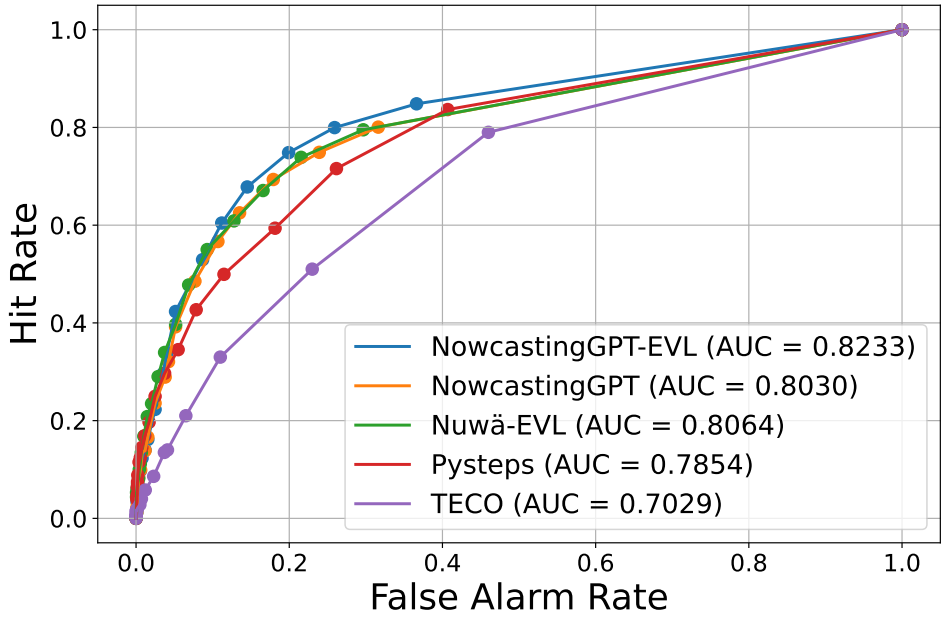


Figure 3.2: Thresholds between 0.5 and 10 for precipitation values are used to define an extreme event. NowcastingGPT-EVL had the highest AUC, outperforming all other baselines.

3.4.2. EXPERIMENTAL RESULTS

We test the performance of the proposed models by using the extreme precipitation test set described in appendix 3.6. Table 3.2 showcases the effectiveness of these methods against a series of metrics that asses the quality and validity of the predictions. The proposed NowcastingGPT-EVL outperforms the other models on the majority of metrics and close second on the rest. The ROC curve in Figure 3.2 demonstrates how NowcastingGPT-EVL outperforms all other methods on extreme event detection at different thresholds. Figure 3.4 illustrates the predicted maps of all considered baselines. While NowcastingGPT presents meaningful predictions over all time steps, Nuwä-EVL deteriorates substantially. Indeed, we believe that when Nuwä-EVL extreme representations get updated by the AT, the VQ-VAE is not able to recognise the extreme latents anymore, which by design are supposed to be fixed, predicting images that do not resemble the ground truth maps semantics. Remarkably, the graphs presented in Appendix 3.6 demonstrate that TECO achieves results on par with other methods, despite having fewer parameters and a more efficient sampling time, and exhibits superior temporal consistency compared to alternative approaches.

	Nuwä-EVL	NowcastingGPT	PySTEPS	TECO	NowcastingGPT-EVL
Number of parameters	772, 832 M	402, 735 M	-	165, 960 M	520, 374 M
Training time	672h	240h	-	155h	264h
Generation time	322.86s	38.90s	9.34s	0.51s	43.10s

Table 3.1: Generation time refers to the time required on average to sample a sequence from the dataset defined in Section 3.6. Training time is computed in terms of GPU hours.

3.5. Conclusion & Discussion

This work proposes NowcastingGPT-EVL, a video prediction model regularized using an EVL regularizer, validating the efficacy of using EVL for nowcasting extreme precipitation events. Our findings reveal that the proposed model outperforms existing methods in various downstream metrics, providing more accurate predictions. The study highlights the importance of addressing data imbalances and the dynamic nature of extreme events in model training. As future work, we aim to assess the prediction capabilities of the different models on an existing and widely used benchmark dataset (e.g., SEVIR [211]). The successful application of NowcastingGPT-EVL underscores the potential of Transformer-based models in enhancing predictive capabilities for critical meteorological forecasting tasks, paving the way for future advancements in the field.

Table 3.2: Each value represents the average and standard deviation over the means and standard deviations of each of the 6 lead times. The description for each metric can be found in appendix 3.6. For statistically meaningful results, we consider 3 different seeds for each entry.

	Nuwä-EVL	NowcastingGPT	PySTEPS	TECO	NowcastingGPT-EVL
PCC (↑)	0.15	0.20 ± 0.002	0.14	0.10 ± 0.002	**0.22** ± 0.002
MSE (↓)	4.85	3.60 ± 0.02	6.22	3.65 ± 0.008	**3.45** ± 0.02
MAE (↓)	1.00	0.72 ± 0.005	0.93	**0.68** ± 0.001	0.69 ± 0.005
CSI(1mm) (↑)	**0.23**	0.21 ± 0.002	0.21	0.07 ± 0.001	0.22 ± 0.002
CSI(2mm) (↑)	**0.13**	0.11 ± 0.001	0.12	0.03 ± 0.001	0.12 ± 0.001
CSI(8mm) (↑)	0.008	0.005 ± 0.0005	**0.01**	0.001 ± 0.0009	0.009 ± 0.0005
FAR(1mm) (↓)	0.61	0.59 ± 0.002	**0.55**	0.69 ± 0.002	0.59 ± 0.002
FAR(2mm) (↓)	0.76	0.71 ± 0.0007	**0.70**	0.78 ± 0.004	0.71 ± 0.0007
FAR(8mm) (↓)	0.85	0.59 ± 0.003	0.89	**0.49** ± 0.006	0.52 ± 0.003
FSS(1km) (↑)	0.35	0.49 ± 0.003	0.32	0.49 ± 0.003	**0.52** ± 0.003
FSS(10km) (↑)	0.42	0.55 ± 0.004	0.41	0.46 ± 0.003	**0.58** ± 0.004
FSS(20km) (↑)	0.48	0.59 ± 0.004	0.47	0.42 ± 0.003	**0.62** ± 0.004
FSS(30km) (↑)	0.52	0.62 ± 0.004	0.51	0.37 ± 0.002	**0.65** ± 0.004

3.6. APPENDIX

DATASET

The reflectivity measurements in the KNMI [212] dataset allows for the estimation of rainfall rates through the application of a Z-R transformation, enabling a nuanced river catchment-level analysis to evaluate the model's effectiveness in real-world scenarios. Ideally, extreme rainfall events are identified based on the distribution of the highest annual rainfall amounts. However, given the limited span of our dataset, which encompasses only 14 years, the dataset provides an insufficient quantity of annual maximums for effective model training and evaluation. Consequently, we have broadened the criteria for what constitutes extreme rainfall. Within this study, an event is classified as extreme if the average precipitation over a three-hour period within a catchment area ranks within the top 1% of all observations recorded from 2008 to 2021. This adjustment allows for a more feasible and statistically sound basis for distinguishing significant precipitation events during the study period. The training dataset consists of 30632 sequences of images with each sequence consisting of 9 images (T-60, T-30, T, T+30, T+60, T+90, T+120, T+150, T+180 minutes) spanning from 2008-2014. The validation dataset consists of 3560 sequences of images with the same sequence length from year 2015-2018. The testing dataset utilised to evaluate the performance of the different models in this study, consists of 357 nationwide extreme events from 2019-2021, corresponding to 3927 events in the catchment regions.

METRICS

The effectiveness of any predictive model is critically assessed through objective metrics that encapsulate its performance capabilities. In our endeavor to evaluate the impact of integrating the EVL regularization, we utilise a comprehensive set of performance metrics:

- *Mean Absolute Error (MAE)*: MAE quantifies the average magnitude of errors in the predictions. It's computed as the mean of the absolute differences between the predicted values and the actual observations, offering a clear and intuitive metric for prediction accuracy.

- *Mean Squared Error (MSE)*: MSE measures the average of the squares of the errors between the predicted and actual values, providing a more sensitive metric that penalizes larger errors more severely than MAE.

- *Pearson Correlation Coefficient (PCC)*: PCC assesses the linear correlation between the predicted and observed datasets, yielding a value between -1 and 1, where 1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 signifies no linear correlation.

- *Critical Success Index (CSI)*: CSI is utilised to evaluate the precision of forecasted events, particularly the successful prediction of specific events. This study examines CSI at two distinct precipitation thresholds: 1mm for light precipitation and 8mm for heavy precipitation, thus catering to varying intensities of rainfall.

- *False Alarm Rate (FAR)*: FAR is calculated as the proportion of false positive predictions relative to the total number of positive forecasts (false positives plus true positives), offering insight into the model's tendency to incorrectly predict events that do not occur.

- *Fractional Skill Score (FSS)*: FSS measures the model's forecast accuracy at specific spatial scales, facilitating an understanding of how well the model performs both locally and over broader areas. In this study, FSS is evaluated at 1km, 10km, 20km and 30km scales to discern the model's effectiveness at varying geographical extents.

While both MAE and MSE loss quantify the quality of the predictions, they are not able to capture the model's capability to detect extreme events. Thus, we make use of a Receiver Operating Characteristic (ROC) curve to assess hit rate detection of extreme precipitation events. The curve is constructed using a set of thresholds that are used to define an event.

BASELINES COMPARISON

Motivated by the overall inefficiency in nowcasting methods, we consider TECO [209] as a point of reference in benchmarking both training and sampling time of Transformer-based nowcasting models. TECO aims to increase sampling efficiency by replacing the common autoregressive prior with a masked token prediction objective, introduced by [213]. Using the discrete tokens from a VQ-VAE, the model learns to predict a randomly generated mask sampled at each timestamp, allowing for orders of magnitude improvement in sampling speed. Moreover, TECO manages to drastically decrease training time by using DropLoss, a trick that allows the model to consider only a subset of the frames that compose the video. Moreover, to be

consistent with the literature, we consider PySTEPS [201], a widely used numerical model for short-term precipitation predictions that achieves remarkable results in nowcasting [214].

NowcastingGPT-EVL description

In this section, we describe the used model. Figure 3.1 illustrates the model architecture. The following subsections describe the three main components: VQ-VAE, Autoregressive Transformer, and Extreme tokens classifier.

Vector Quantized Variational AutoEncoder

The Vector Quantized Variational AutoEncoder (VQVAE) [186] introduces a novel approach by utilizing Vector Quantization to encode inputs into discrete latent representations, moving away from continuous feature representations. This method is effective in capturing the complex, multi-dimensional features of data. VQVAE operates on an encoder-decoder framework with a discrete codebook, where the encoder compresses input data into a discrete set of codes, preserving essential features through a reduction in spatial dimensions and an increase in feature channels. The decoder then reconstructs the input from these codes, aiming for a close approximation to the original, thereby enabling efficient and structured data representation suitable for tasks like image reconstruction. The encoder consists of 5 downsampling layers each containing 2 ResNet blocks, thus reducing the spatial dimension of the input to the following resolutions: $128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8$. Furthermore, the last stage of the encoder includes an attention block used to capture the relationships between features before the quantization step. In order to obtain the reconstructed image from the discrete codes, we use a decoder that mirrors the structure of the encoder.

To facilitate the training of the VQVAE model, a set of distinct loss functions are harnessed and subjected to optimization. These loss functions encompass the reconstruction loss, the commitment loss, and the perceptual loss.

$$\mathcal{L}(E, D, \mathcal{Z}) = \|x - \hat{x}\|_2^2 + \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 \quad + \|\text{sg}\left[z_{\mathbf{q}}\right] - E(x)\|_2^2 + \mathcal{L}_{\text{perceptual}}(x, \hat{x}), \tag{3.2}$$

where $\hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z}$ represents the encoded image while $\hat{x} = D(z_q)$ is the reconstructed image using $z_{\mathbf{q}}$. We obtain $z_{\mathbf{q}}$ using an element-wise quantization $q(.)$ of each spatial code $\hat{z}_{ij} \in \mathbb{R}^{n_z}$ given by

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\arg\min_{z_k \in \mathcal{Z}} \left\| \hat{z}_{ij} - z_k \right\| \right) \in \mathbb{R}^{h \times w \times n_z}.$$

Autoregressive Transformer

In order to model the dynamics between consecutive precipitation maps, we use an Autoregressive Transformer. For training the model, we utilise the ground truth precipitation maps which are quantized into $\mathbf{z}_q = q(\mathbf{E}(x))$, generating a sequence $\mathbf{s} \in \{0, \ldots, |\mathbf{Z}| - 1\}^{h \times w}$, corresponding to the respective indices of the

VQVAE codebook. Subsequently, these indices are transformed into continuous vector representations using an embedder. In order to provide sequence order information to the Transformer, the representations are augmented with positional embeddings. These are then processed by the Transformer, which outputs the logits generated by the head module. These logits represent the probability of using a specific token and are used to compute a cross-entropy loss. The loss compares the predicted probabilities given by the model with the actual token probabilities:

$$\mathbb{L}_{\text{Transformer}} = \mathbb{E}_{x\sim p(x)}[-\log \prod_{i=1}^{N} p(\mathbf{s}_i|\mathbf{s}_{<i})] \tag{3.3}$$

which, given a sequence of indices $\mathbf{s}_{<i}$, the Transformer is trained to predict the distribution of the consecutive indices $\mathbf{s}_i$. The AT employs a causal attention mechanism, where the non-causal entries of QK^T, those below the diagonal of the attention matrix, are set to $-\infty$. As a result, the attention mechanism accesses only previously seen or current tokens when predicting the next one in a sequence, enabling efficient and context-sensitive output production. We use the architecture described above to define our ablation model NowcastingGPT.

The Autoregressive Transformer for NowcastingGPT-EVL has the EVL loss function incorporated in it so the overall loss function for the AR transformer is given as:

$$\mathcal{L}_{\text{Transformer(NowcastingGPT-EVL)}} = \mathcal{L}_{\text{Transformer}} + \lambda[\text{EVL}(u_t, v_t)]. \tag{3.4}$$

The value of λ in the equation above is chosen as 0.5.

BINARY CLASSIFIER

For the classification of the tokens into extreme or non-extreme, a transformer is incorporated along with the auto-regressive transformer. The input to this transformer are the sequence of tokens that are generated from the auto-regressive transformer during its training phase. The model has 6 layers, 1024 embedding dimension and, a total number of 8 heads. The transformer is trained using a standard binary cross entropy loss function where, the ground truth labels v_t are calculated on the basis of averaged precipitation over a threshold of 5mm. In this way, all the tokens corresponding to an extreme/non-extreme event get classified along with the training of the auto-regressive transformer. The classifier generates logits for the two classes (extreme and, non-extreme) which are then passed through a softmax layer to generate probabilities. These probabilities act as the input to the EVL loss function mentioned in equation (3.1) for the term u_t. The values for β_0 and β_1 are taken as 0.95 and 0.05 respectively since, top 5% of the events are considered as extreme events. The value of γ for EVL was set to 1, as this setting demonstrated optimal performance.

MATHEMATICAL PROOF OF THE EVL LOSS FUNCTION

As mentioned in [215], if there is a sequence of independent and identically distributed (i.i.d) random variables as $X_1, X_2, \ldots, X_n$, having marginal distribution function

F. It is natural to regard as extreme events those of the X_i that exceed some high threshold u. Denoting an arbitrary term in the X_i sequence by X, it follows that a description of the stochastic behaviourof extreme events is given by the conditional probability:

$$\Pr\{X > u + y \mid X > u\} = \frac{1 - F(u + y)}{1 - F(u)}, \quad y > 0. \tag{3.5}$$

Starting from the L.H.S we have:

$$\Pr\{X > u + y \mid X > u\},$$

and using the formula $P(x \mid y) = \frac{P(x,y)}{P(y)}$:

$$\begin{aligned}\Pr\{X > u + y \mid X > u\} &= \frac{P(X > u + y, X > u)}{P(X > u)} \\ &= \frac{P(X > u + y)}{P(X > u)}.\end{aligned}$$

Applying the formula, $P(X > x) = 1 - F(x)$ we get,

$$= \frac{1 - F(u + y)}{1 - F(u)}.$$

If the parent distribution F was known to us then the distribution of threshold exceedances in equation 3.5 would also be known, however, that is not the case. [215] suggests the application of Extreme Value Theory (EVT) for the approximation of the distribution of maxima of long sequences when the parent population function (distribution) F is unknown. For the sequence of R.Vs mentioned above (with common distribution function F), we use maximum order statistics to characterize extremes :

$$M_n = \max\{X_1, X_2, X_3, \ldots X_n\}, \xrightarrow{P} x^*, n \to \infty. \tag{3.6}$$

where $\xrightarrow{P}$ denotes convergence in probability and, x^* denotes the right end point which is $x^* = \sup\{x : F(x) < 1\}$ Therefore, for a large n we have :

$$P(\max(X_1, X_2, \ldots, X_n) \leqslant x) = Pr(X_1 \leqslant x, X_2 \leqslant x, X_3 \leqslant x, \ldots X_n \leqslant x), \tag{3.7}$$

since, they are i.i.d we can also write equation 3.7 as,

$$P(\max(X_1, X_2, \ldots, X_n) \leqslant x) = [\Pr(X \leqslant x)]^n = [\mathrm{F}(x)]^n.$$

Hence, from

$$[F(x)]^n \to 0 \text{ for } x < x^*$$
$$[F(x)]^n \to 1 \text{ for } x \geqslant x^*,$$

it can be said said that $[F(x)]^n$ is a degenerate function as it converges to a single point when n becomes sufficiently large. To mitigate this, EVT suggests that for

a sequence of constants $a_n > 0$ and real b_n there is a non-degenerate distribution function G stated as:

$$\lim_{n\to\infty} [F(a_n x + b_n)]^n = G(x), \tag{3.8}$$

where $G(x)$ is the Generalised Extreme Value distribution function (GEV). The GEV is given by:

$$G(x) = \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \tag{3.9}$$

where μ is the location parameter, σ is the scale parameter and ξ is the shape parameter. Also, equation (3.8) can be written as:

$$\begin{aligned} & [F(a_n x + b_n)]^n \approx G(x) \\ \implies & [F(x)]^n \approx G\{(x-b_n)/a_n\} \\ \implies & [F(x)]^n = G^*(x), \end{aligned}$$

where G^* is another member of the GEV family. [215] mentions that if equation (3.8) allows the approximation of $[F(a_n x + b_n)]^n$ by a member of the GEV family for large n, then $[F(x)]^n$ can also be approximated using a different member of the GEV family $(G^*(x))$ which has the same definition as mentioned in 3.9 but with different values of μ, σ and ξ. Therefore, we can then write :

$$[F(x)]^n \approx \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}. \tag{3.10}$$

Taking natural logarithm on both sides,

$$n(\ln F(x)) \approx -\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}.$$

For large values of x, a Taylor expansion implies that,

$$\ln F(x) \approx -\{1-F(x)\}.$$

Substituting this in the above equation we get,

$$1-F(x) \approx \frac{1}{n}\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}. \tag{3.11}$$

Now, we substitute the above result obtained in the R.H.S of equation (3.5) for a large u and $y > 0$ as,

$$1-F(u) \approx \frac{1}{n}\left[1+\xi\left(\frac{u-\mu}{\sigma}\right)\right]^{-1/\xi}$$

and,

$$1 - F(u+y) \approx \frac{1}{n}\left[1 + \xi\left(\frac{u+y-\mu}{\sigma}\right)\right]^{-1/\xi}.$$

Therefore, we can write equation (3.5) as:

$$\begin{aligned}\Pr\{X > u+y \mid X > u\} &\approx \frac{n^{-1}[1+\xi(u+y-\mu)/\sigma]^{-1/\xi}}{n^{-1}[1+\xi(u-\mu)/\sigma]^{-1/\xi}} \\ &= \left[1 + \frac{\xi y/\sigma}{1+\xi(u-\mu)/\sigma}\right]^{-1/\xi} \\ &= \left[1 + \frac{\xi y}{\tilde{\sigma}}\right]^{-1/\xi},\end{aligned} \tag{3.12}$$

where $\tilde{\sigma} = \sigma + \xi(u-\mu)$. This distribution function is known as the Generalised Pareto Distribution (GPD) function which helps in modeling observations over a large enough threshold u (Peaks Over Threshold method - POT) and is written formally as :

$$H(y) = 1 - \left(1 + \frac{\xi y}{\tilde{\sigma}}\right)^{-1/\xi}, \tag{3.13}$$

defined on $\{y : y > 0 \text{ and } (1 + \xi y/\tilde{\sigma}) > 0\}$, where

$$\tilde{\sigma} = \sigma + \xi(u-\mu).$$

According to [215], the above relation in equation 3.12 implies that, if block maxima have approximating distribution G, then threshold excesses have a corresponding approximate distribution within the GPD family (H). Also, the parameters of GPD can be uniquely determined by those of the associated GEV distribution of block maxima. Moreover, the GEV distribution function and the GPD distribution function are related to each other since they have the same shape parameter ξ so we can derive a rough mathematical relation between these two distribution functions as:

$$H(y) = 1 + \ln(G(y)), \tag{3.14}$$

for some location (μ) and shape ($\sigma, \tilde{\sigma}$) parameters. This relationship is also mentioned in the paper [216] which utilises EVT and Value-at-Risk for relative performance of stock market returns in emerging markets. We can rewrite equation (3.5) with the help of the derived results in equations (3.12) and (3.13) as:

$$\begin{aligned}\frac{1-F(u+y)}{1-F(u)} = \left[1 + \frac{\xi y}{\tilde{\sigma}}\right]^{-1/\xi} &\implies \frac{1-F(u+y)}{1-F(u)} = 1 - H(y) \\ &\implies 1 - F(u+y) \approx (1-F(u))(1-H(y)).\end{aligned} \tag{3.15}$$

This equation is the main equation for the tail approximation of observations exceeding a threshold u and matches with the tail approximation equation mentioned in [217] as:

$$1 - F(x) \approx (1 - F(t)) \left\{ 1 - H_\xi \left(\frac{x - t}{f(t)} \right) \right\}, x > t, \tag{3.16}$$

where H_ξ is the GPD function with the shape parameter ξ. Therefore, we use the result derived in equation (3.15) to derive the weights of the EVL loss function mentioned in the paper [203]. However, the authors utilise the GEV distribution function to define the underlying distribution of the time series data used in the paper. The goal of the paper is to predict outputs $Y_{T:T+K}$ in the future given the observations ($X_{1:T}, Y_{1:T}$ and future inputs $X_{T:T+K}$. For the sake of convenience, the authors define $X_{1:T} = [x_1, \cdots, x_T]$ and $Y_{1:T} = [y_1, \cdots, y_T]$ to denote the general input and output sequences without referring to specific sequences. Therefore, for T random variables $y_1, \cdots, y_T$ i.i.d sampled from a distribution F_Y, the distribution of the maximum is realised using EVT as :

$$\lim_{T \to \infty} P\left\{ \max\left(y_1, \cdots, y_T\right) \leq y \right\} = \lim_{T \to \infty} F^T(y) = G(y), \tag{3.17}$$

for some linear transformation where $G(y)$ is GEV distribution function. We can observe that equation (3.8) and equation (3.17) have the same meaning (but with different variables in their definitions). Moreover, the authors define the GEV function as:

$$G(y) = \begin{cases} \exp\left(- \left(1 - \frac{1}{\gamma} y \right)^\gamma \right), & \gamma \neq 0, 1 - \frac{1}{\gamma} y > 0 \\ \exp\left(-e^{-y} \right), & \gamma = 0, \end{cases} \tag{3.18}$$

where γ is known as the extreme value index (the shape parameter) with condition $\gamma \neq 0$. It can also be observed that the definition of GEV function in equation (3.18) is similar to the definition mentioned in equation (3.9) but with $\xi = -\frac{1}{\gamma}$, $\mu = 0$ and, $\sigma = 1$. For modeling the tail distribution of the corresponding time-series data, they use equation (3.16) but as mentioned before, rather than using the GPD function they use the GEV distribution function to model the tail approximation. Therefore, we substitute the relationship mentioned in equation (3.14) as $-ln(G(y)) = 1 - H(y)$ in equation (3.16) and get the following result:

$$1 - F(y) \approx (1 - F(\xi)) \left[- \ln G \left(\frac{y - \xi}{f(\xi)} \right) \right], y > \xi, \tag{3.19}$$

where ξ is the threshold and, $f(\xi)$ is a scale function as mentioned in the paper [203]. Also, the authors define an extreme indicator sequence $V_{1:T} = [v_1, \cdots, v_T]$ as:

$$v_t = \begin{cases} 1 & y_t > \xi \\ 0 & y_t \leqslant \xi, \end{cases} \tag{3.20}$$

where ξ is the threshold. For time step t if $v_t = 0$ then the output y_t is considered as a 'normal event' and if $v_t = 1$ then y_t is considered as an 'extreme event'. The authors also mention a hard approximation for the term $(\frac{y-\xi}{f(\xi)})$ in equation (3.19) as u_t which is the predicted indicator by the neural network used by them in their experiment. This can be interpreted as a normalization which restricts the values of output y, above and below the threshold ξ between $[-1, 1]$. Therefore, considering this to be true, we can rewrite equation (3.19) as:

$$1 - F(y) \approx (1 - F(\xi)) \left[- \ln G(u_t) \right], \tag{3.21}$$

Substituting the definition of GEV in equation (3.18) into the above equation (3.21) we obtain:

$$1 - F(y) \approx (1 - F(\xi)) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}. \tag{3.22}$$

The term $1 - F(\xi)$ can be approximated as:

$$1 - F(\xi) = \Pr(y > \xi) \implies 1 - F(\xi) = \Pr(v_t = 1), \tag{3.23}$$

where $\Pr(v_t = 1)$ is the proportion of extreme events in the dataset. Therefore, we can rewrite equation (3.22) with the above substitution as:

$$1 - F(y) \approx \Pr(v_t = 1) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}. \tag{3.24}$$

This tail approximation is incorporated in the terms of the standard Cross Entropy (CE) function as weights to define the main EVL loss function mentioned in paper [203]. However, the authors in paper [203] define the weight as:

$$1 - F(y) \approx (1 - \Pr(v_t = 1)) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}. \tag{3.25}$$

Upon simplifying the term $(1 - \Pr(v_t = 1)$ we get:

$$\begin{aligned} & 1 - \Pr\left(v_t = 1\right) \\ = & \Pr\left(v_t = 0\right) \\ = & \Pr(y \leqslant \xi) \\ = & F(\xi), \end{aligned} \tag{3.26}$$

so we get the expression $1 - F(y) \approx (F(\xi)) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}$ which is not in congruence with the main tail approximation in equation (3.19) as shown by [203]. Moreover, research by [218] show similar weight derivations for the EVL loss function as it has been derived in equation (3.24). Therefore, applying the weights derived in equation (3.24) to the standard BCE loss function, we get:

$$\begin{aligned}\mathrm{EVL}\left(u_t, v_t\right) = &- \Pr(v_t = 1)\left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log\left(u_t\right) \\ &- \Pr(v_t = 0)\left[1 - \frac{1 - u_t}{\gamma}\right]^{\gamma} (1 - v_t) \log\left(1 - u_t\right),\end{aligned} \tag{3.27}$$

where the standard BCE loss function for a binary classification task is given by:

$$\begin{aligned}\mathrm{BCE}\left(u_t, v_t\right) = &- v_t \log\left(u_t\right) \\ &- (1 - v_t) \log\left(1 - u_t\right).\end{aligned} \tag{3.28}$$

ADDITIONAL RESULTS

This section supports the findings in Section 3.4.2 with a series of qualitative results that provide a different perspective for assessing the quality of the results. Thus, the emphasis is on the quality difference between lead times on the extreme precipitation events dataset. Figure 3.3 aims to provide visualizations of the predicted lead times as a visual signal on the quality of the predictions. On the other hand, Figure 3.4 to Figure 3.16 provide additional analysis on the performance of the proposed models on the metrics presented in Section 3.4.2

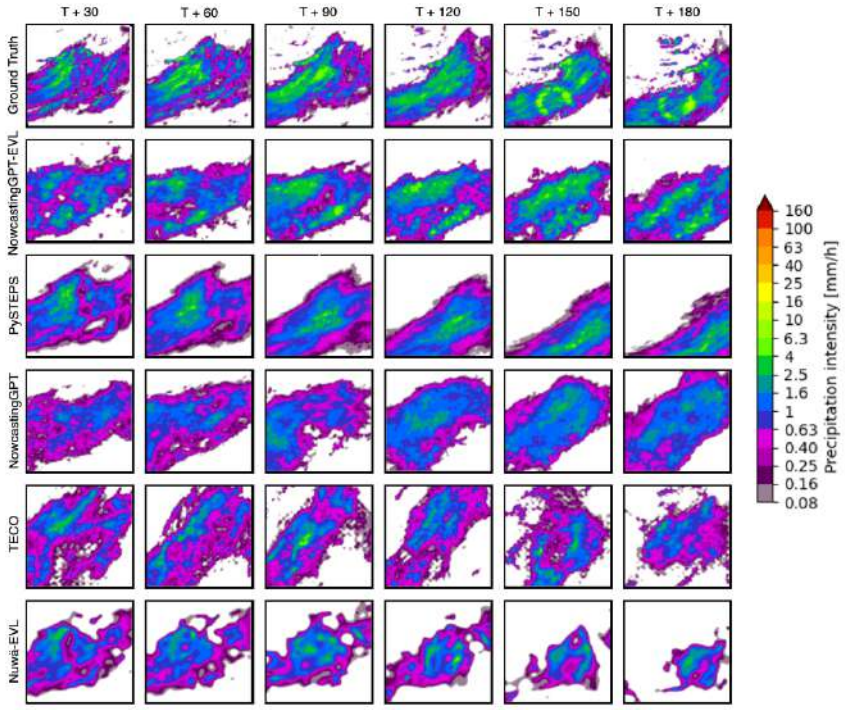


Figure 3.3: The generation is conditioned on 3 previous timestamps with the task to predict the next 6 lead times. There is a gap of 30 minutes between each timestamp. Images are upsampled to 256×256 pixels.

Autoregressive Transformer: The AT architecture aims to model the dynamics between consecutive precipitation maps. During training, the ground truth precipitation maps are quantized into $\mathbf{z}_q = q(\mathbf{E}(x))$, producing a sequence $\mathbf{s} \in \{0, \dots, |\mathbf{Z}| - 1\}^{h \times w}$, representing VQ-GAN codebook indices. These indices are transformed into continuous vectors by an embedder and augmented with positional embeddings to provide order information for the transformer. The transformer then processes these vectors, with the head module refining the output into logits, which represents the probability of using a specific token. These logits are used to compute

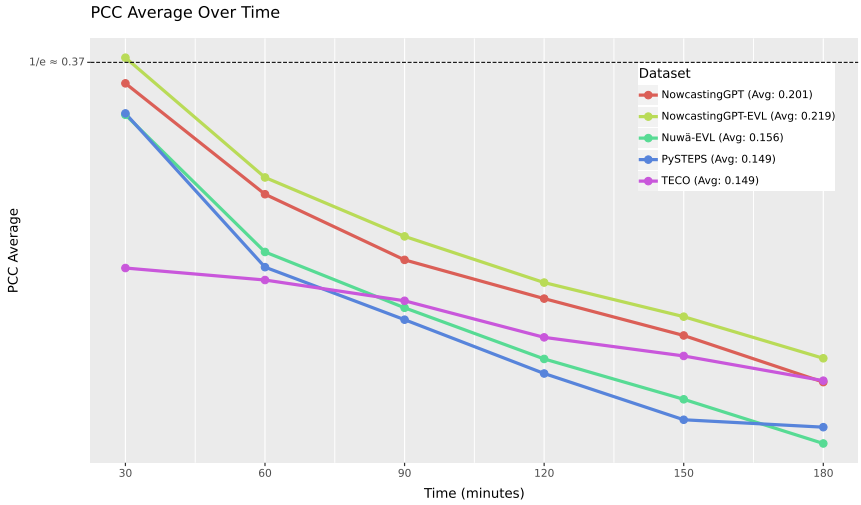


Figure 3.4: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. NowcastingGPT-EVL outperforms all other models.

a cross-entropy loss that compares predicted tokens probabilities with the actual tokens:

$$\mathbb{L}_{\text{Transformer}} = \mathbb{E}_{x\sim p(x)}[-\log \prod_{i=1}^{N} p(\mathbf{s}_i|\mathbf{s}_{<i})] \tag{3.29}$$

which, given a sequence of indices $\mathbf{s}_{<i}$, the transformer is trained to predict the distribution of the consecutive indices $\mathbf{s}_i$. The AT employs a causal attention mechanism, where the non-causal entries of QK^T, those below the diagonal of the attention matrix, are set to $-\infty$. As a result, the attention mechanism accesses only previously seen or current tokens when predicting the next one in a sequence, enabling efficient and context-sensitive output production.

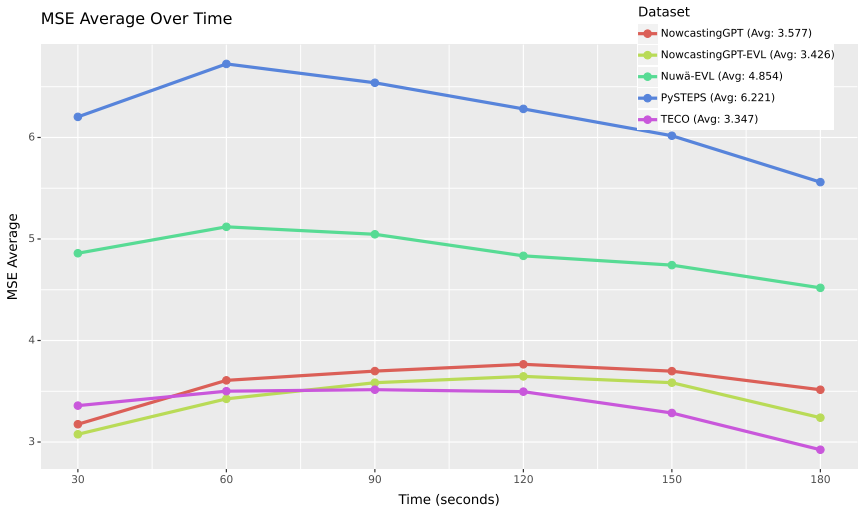


Figure 3.5: Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. NowcastingGPT-EVL and TECO outperforms all other models for bigger lead times.

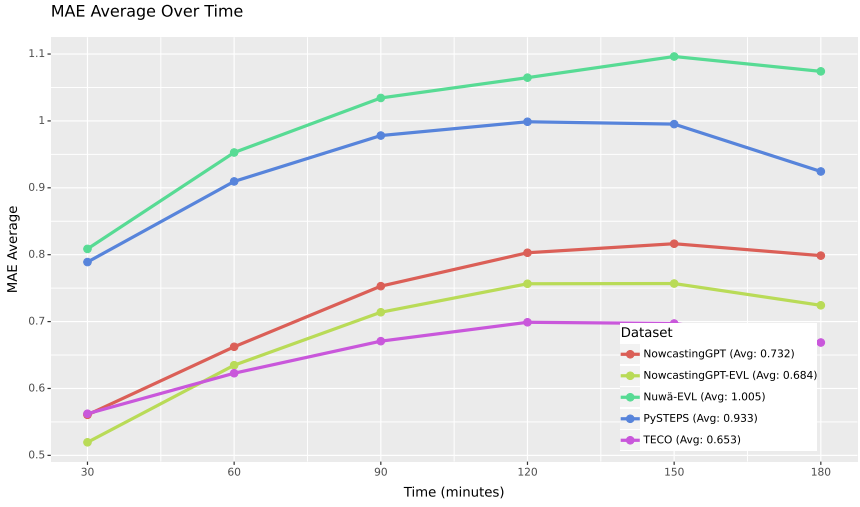


Figure 3.6: Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. TECO outperforms all other models.

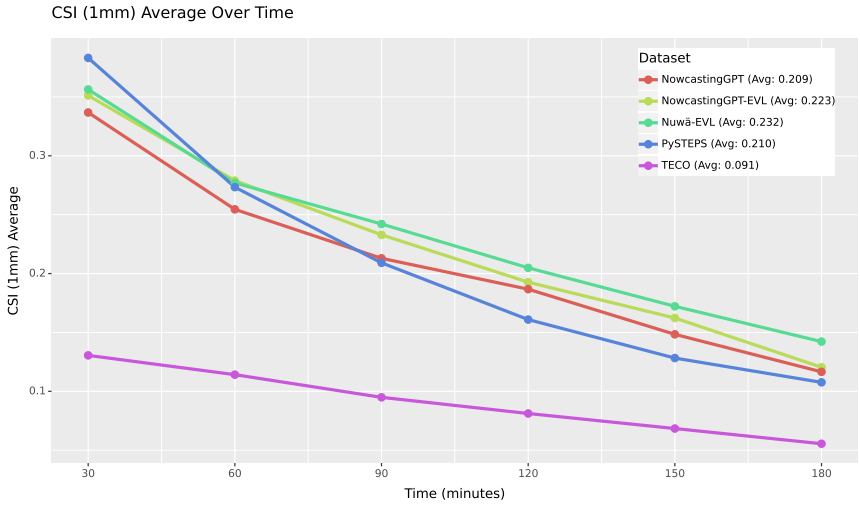


Figure 3.7: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. Nuwä-EVL and NowcastingGPT-EVL outperform the rest of the models but decay quickly.

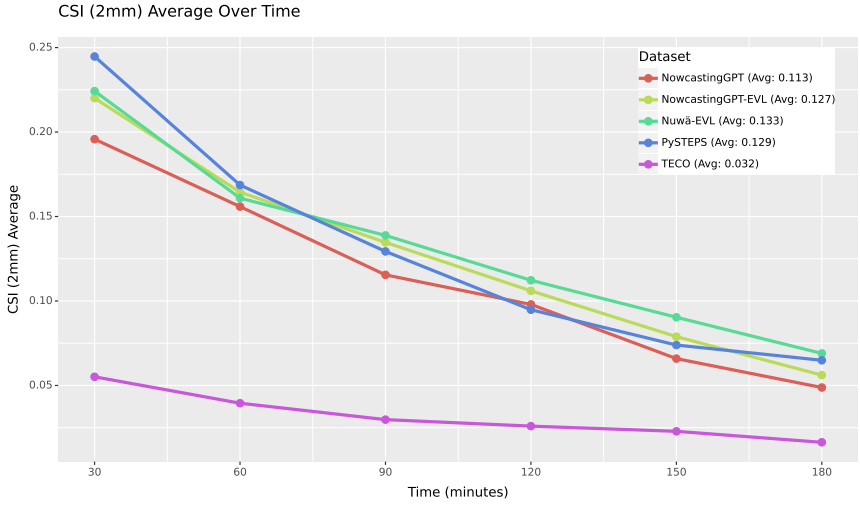


Figure 3.8: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. Nuwä-EVL outperforms the rest of the models.

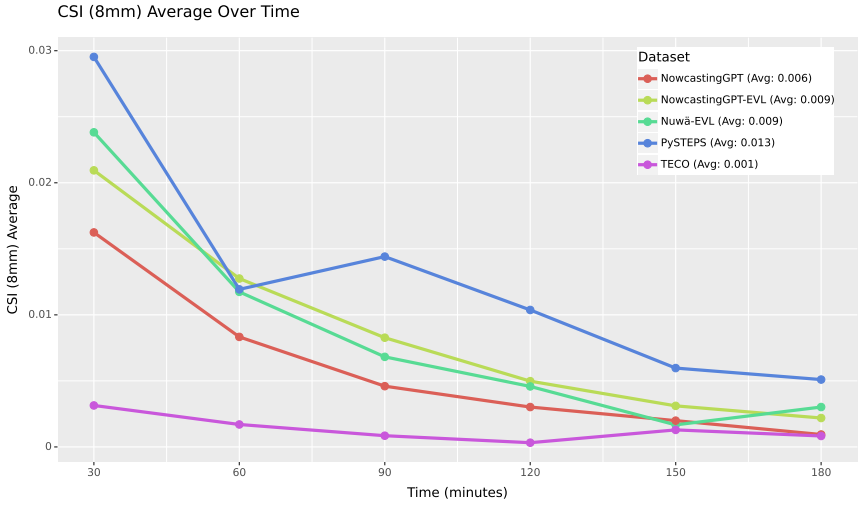


Figure 3.9: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. PySTEPS outperforms the rest of the models.

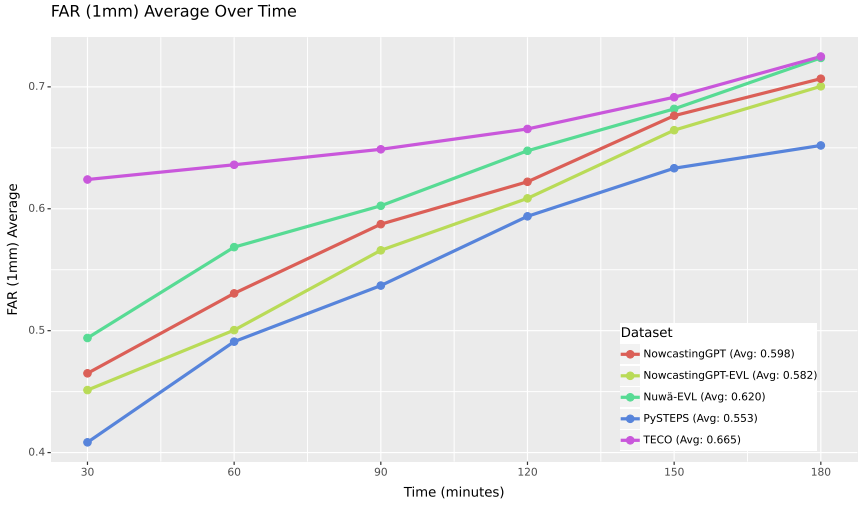


Figure 3.10: Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. PySTEPS outperforms the rest of the models.

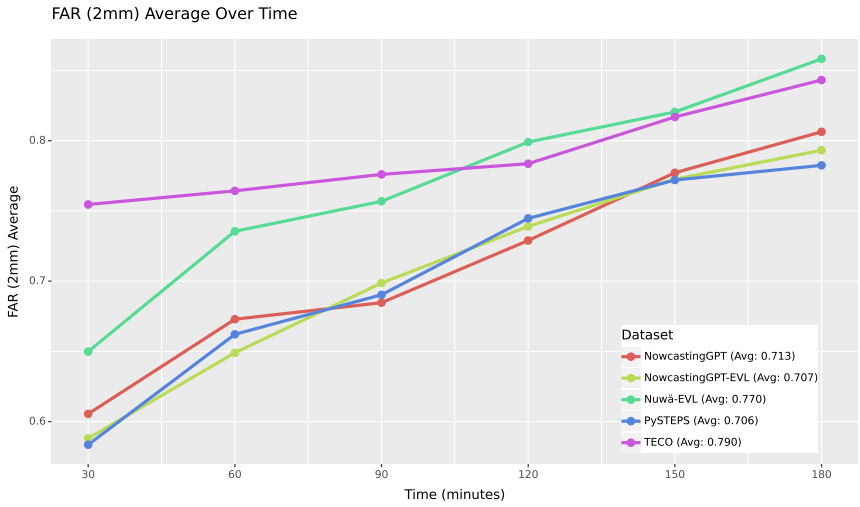


Figure 3.11: Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. NowcastingGPT-EVL outperforms the rest of the models.

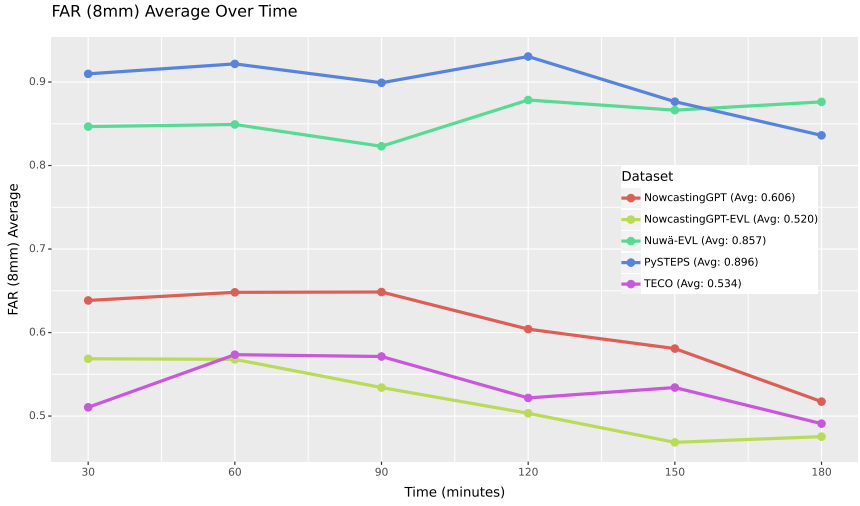


Figure 3.12: Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. NowcastingGPT-EVL and TECO outperform the rest of the models.

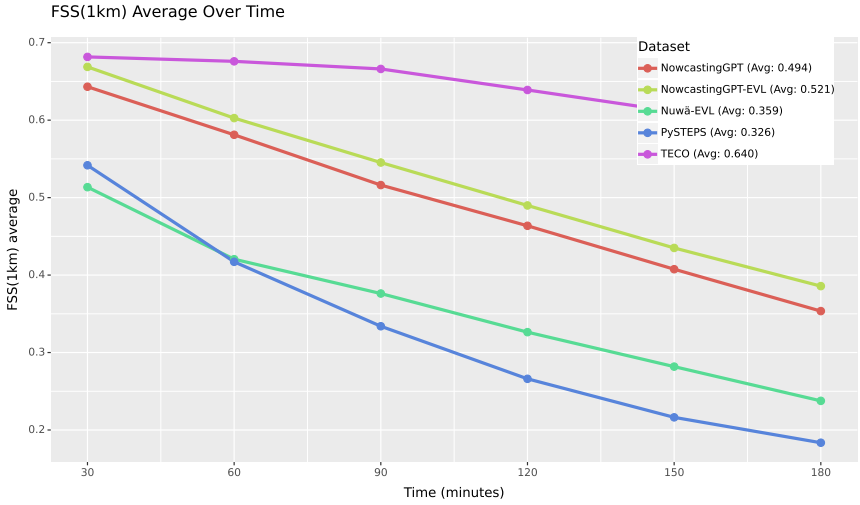


Figure 3.13: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. TECO outperforms the rest of the models.

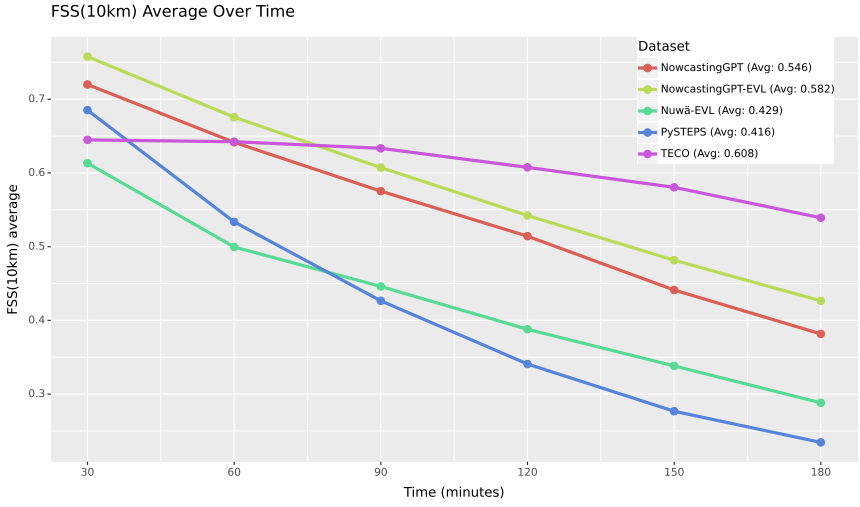


Figure 3.14: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. TECO outperforms the rest of the models on higher lead times while NowcastingGPT and NowcastingGPT-EVL perform better on lower lead times.

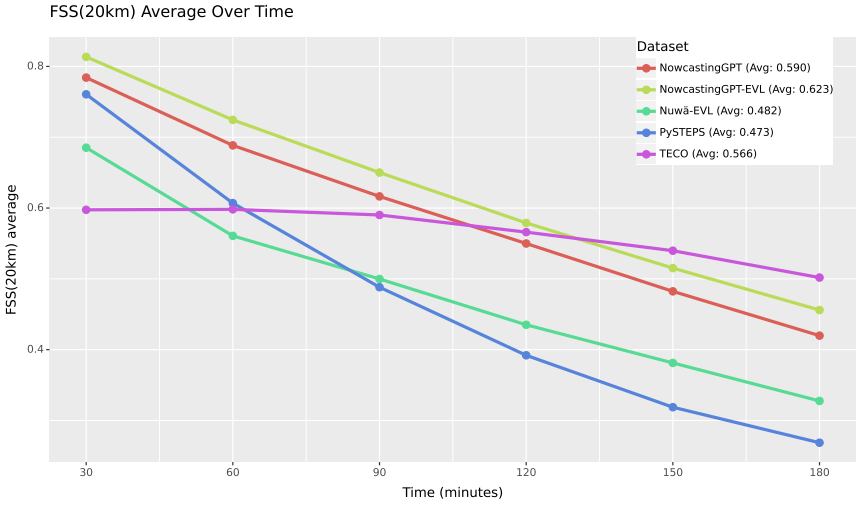


Figure 3.15: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. NowcastingGPT-EVL outperforms the rest of the models.

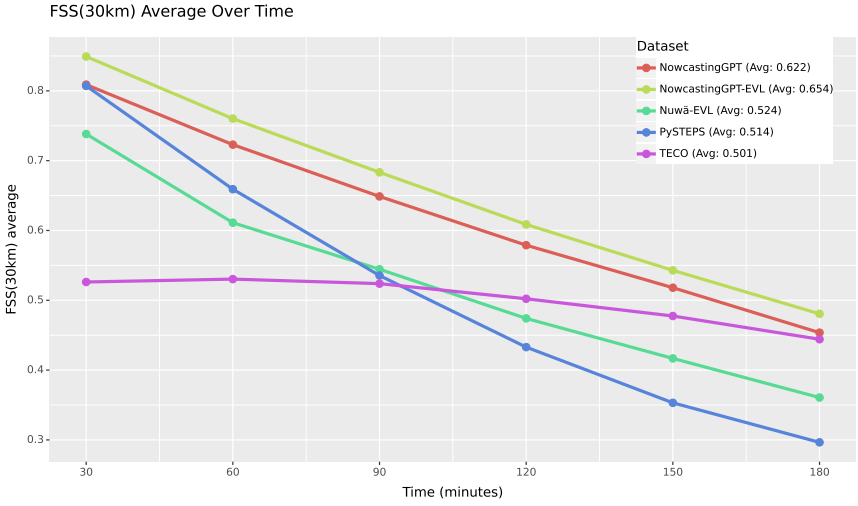


Figure 3.16: Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. NowcastingGPT-EVL outperforms the rest of the models.

4

PRECIPITATION NOWCASTING USING PHYSICS-INFORMED DISCRIMINATOR GENERATIVE MODELS

Published at the European Conference on Signal Processing (EUSIPCO 2024)

Junzhe Yin*
Delft University of Technology, NL
junzhe.yin@student.tudelft.nl

Cristian Meo*
Delft University of Technology, NL

Ankush Roy
Delft University of Technology, NL

Zeineb Bou Cher
Delft University of Technology, NL

Yanbo Wang
Delft University of Technology, NL

Ruben Imhoff
Deltares, NL

Remko Uijlenhoet
Deltares, NL

Justin Dauwels
Delft University of Technology, NL

Nowcasting leverages real-time atmospheric conditions to forecast weather over short periods. State-of-the-art models, including PySTEPS, encounter difficulties in accurately forecasting extreme weather events because of their unpredictable distribution patterns. In this study, we design a physics-informed neural network to perform precipitation nowcasting using the precipitation and meteorological data from the Royal Netherlands Meteorological Institute (KNMI). This model draws inspiration from the novel Physics-Informed Discriminator GAN (PID-GAN) formulation, directly integrating physics-based supervision within the adversarial learning framework. The proposed model adopts a GAN structure, featuring a Vector Quantization Generative Adversarial Network (VQ-GAN) and a Transformer as the generator, with a temporal discriminator serving as the discriminator. Our findings demonstrate that the PID-GAN model outperforms numerical and SOTA deep generative models in terms of precipitation nowcasting downstream metrics.

4.1. INTRODUCTION

[1] More frequent global extreme precipitation has led to severe flooding, soil erosion, loss of agricultural productivity, and heightened health risks[219]. Traditional Numerical Weather Prediction (NWP) models, though comprehensive, face challenges in short-term forecasting of rainfall due to their high computational requirements and too low spatial-temporal resolution [200, 214]. Precipitation nowcasting techniques aim to provide accurate forecasts of upcoming precipitation events within six hours for local regions, reducing response time and efforts to handle extreme weather events [202, 220, 221]. Radar extrapolation methods, such as PySteps, leverage real-time data and use optical flow and statistical analysis techniques to improve weather prediction accuracy [201]. However, while statistical nowcasting methods eliminate the need for historical data, they do not explicitly account for the growth and decay processes of convective rainfall, limiting their usefulness for severe rainfall events. Due to limitations in radar extrapolation methods, there has been a shift toward deep learning models [202, 222], which have shown promise in weather prediction by capturing complex spatio-temporal patterns without relying on traditional data assimilation techniques.

Deep generative models like Generative Adversarial Networks (GAN) [15] and Variational Autoencoder (VAE) [14] have excelled in precipitation nowcasting, offering more accurate and realistic forecasts by modelling the distribution underlying the precipitation dynamics [202, 222, 223]. Nevertheless, these models have many limitations, for instance, they are not able to produce consistent prediction over medium- to long-term horizons and struggle with extreme event modeling and prediction. Moreover, a notable limitation of these models is their tendency to overlook fundamental physical laws, resulting in predictions that may not always align with them, particularly in scenarios not encountered during training. Physics-informed machine learning (PIML) [224] has emerged as a promising solution, enhancing weather and climate forecasting by improving physical consistency [225]. In this paper, we propose a novel PIML model that integrates deep generative models

[1] Accepted at European conference on signal processing (EUSIPCO) 2024

with physical priors of precipitation, aiming to produce accurate and physically consistent precipitation forecasts.

4.1.1. Dataset

This paper explores precipitation nowcasting in the Netherlands, using weather radar data from the Royal Netherlands Meteorological Institute (KNMI) for 2008-2021 [222], along with hourly meteorological data from Automatic Weather Stations (AWS) [212] and ERA5 reanalyses [226]. The meteorological data from AWS contains hourly observations of wind speed, wind direction, air temperature, dew point temperature and global radiation. ERA5 offers hourly estimates for a wide variety of quantities related to the atmosphere, ocean waves, and the land surface [226]. We use radar precipitation maps with a 5-minute temporal resolution and a 1 km spatial resolution. Our study focuses on a 256×256 pixel area covering most of the country and 12 Dutch catchment regions, as detailed in [[214], Fig. 1]. Following [222], an event is classified as extreme if its average rainfall over three hours ranks in the top 1% of all recorded events between 2008 and 2021. This threshold is defined as exceeding 5 mm per 3 hours, as shown in [[222], Table. 1]. To match the temporal resolution of AWS and ERA5 meteorological data with the used radar dataset, we apply cubic interpolation to estimate half-hour intervals of the latter (e.g., $T-60$, $T-30$, to $T+180$ minutes) from the original hourly data points (e.g., $T-60$, T, $T+60$, $T+120$, $T+180$ minutes). To address the limited spatial coverage of the AWS and ERA5 reanalyses datasets, we applied kriging interpolation using the PyKrige package [227], achieving a spatial resolution compatible with our radar data. Kriging uses spatial autocorrelation and variance to accurately estimate values in unmeasured locations, creating maps that align with the resolutions of the radar map [228].

4.1.2. Problem Formulation

In the last few years, precipitation nowcasting using deep learning models have been cast as a video prediction problem [222], where given an input spatio-temporal sequence of N frames $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{N \times H \times W \times C}$, where H, W denote the spatial resolution and C represents the image channels or the different types of measurements (e.g., radar maps, humidity maps, etc). The goal is to predict the next M precipitation maps $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{M \times H \times W \times 1}$ and classify extreme events based on the defined threshold. Throughout this work we use $N = 3$ and $M = 6$.

4.2. Related Works

Generative Adversarial Networks, as the current state-of-the-art in various fields, have seen many adaptable variations in recent years, demonstrating their flexibility. They serve as key architectures in precipitation nowcasting tasks. For instance, the Adversarial Extrapolation Neural Net (AENN) [223], a variation of GAN with 2 discriminators, has shown superior performance in weather radar echo extrapolation than NWP approaches. In [202], a GAN utilizing temporal and spatial discriminators was proposed. This advancement underscores the potential of GANs in enhancing

the accuracy of precipitation nowcasting through sophisticated data generation techniques.

Despite their advancements, these models face persistent challenges, such as delivering consistently clear and precise forecasts and adequately generalising across diverse weather conditions, particularly those underrepresented in training datasets. A significant obstacle faced by sophisticated neural networks such as AENN is their lack of interpretability. A critical hurdle with advanced neural networks lies in their interpretability. The uninterpretable reasoning behind their predictions complicates the understanding of their decision-making processes, making it difficult for experts to fully trust and utilise these models. Moreover, these systems must adhere to the fundamental physical principles that govern meteorological phenomena, ensuring that their predictive capabilities do not compromise physical accuracy. Maintaining a balance between physical accuracy and predictive performance represents a significant challenge, underscoring the demand for innovation in meteorological forecasting. This balance is crucial for developing reliable models that can accurately predict weather patterns while adhering to the fundamental principles of meteorology. Recently, PID-GAN, a physics-informed discriminator GAN formulation that does not suffer from an imbalance of gradient flow compared to physics-informed neural networks (PINN) [229], was introduced in [230]. In this work, we adopt the PID-GAN framework to integrate physical constraints into our model, ensuring the generation of physically coherent precipitation forecasts. This includes the incorporation of moisture conservation equations, as outlined in [231], to accurately capture the dynamics of precipitation.

4.3. METHODOLOGY

In this section, we define and describe the proposed PID-GAN architecture used in this work. Moreover, we describe the physics-based loss function used to inject the moisture conservation equation information into the PID-GAN architecture.

4.3.1. PID-GAN: MODEL ARCHITECTURE

This section describes the backbone architecture used for the proposed PID-GAN model, which is inspired by the successful implementation detailed in [202]. Our design adopts this proven structure, incorporating a generator alongside spatial and temporal discriminators, due to its demonstrated effectiveness in capturing complex spatio-temporal relationships in data. This approach ensures a robust framework capable of handling the intricate dynamics of precipitation nowcasting by efficiently learning from both the spatial patterns and temporal sequences inherent in meteorological data.

GENERATOR

The generator in our model incorporates a deep generative model with two components: Vector Quantization Generative Adversarial Network (VQ-GAN) [208], that learns the mapping between precipitation maps and discrete tokens, and an

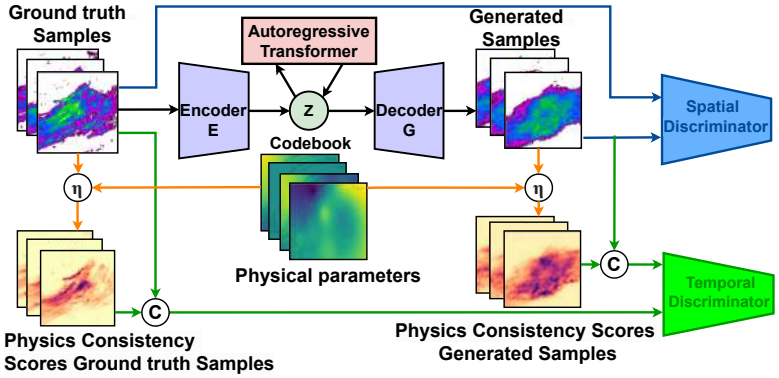


Figure 4.1: C stands for concatenation and η represent $\eta_k = e^{-\lambda \mathcal{R}^{(k)}(x,\hat{x})}$, refering to the equation of the physics consistency score.

Autoregressive Transformer (AT) [5] that models the dynamics between tokens of consecutive timesteps.

VQ-GAN: The architecture of the VQ-GAN model consists of a CNN encoder (E) and decoder (G), a codebook ($\mathcal{Z}$), which define a VQ-VAE [186] and a patch-based discriminator [232], indicated as spatial discriminator (D). The VQ-VAE is trained by optimising:

$$\begin{aligned}\mathcal{L}_{\text{vq-vae}} = &\|x - \hat{x}\|_1 + \|sg[E(x)] - z_q\|_2^2 \\ &+ \|sg[z_q] - E(x)\|_2^2 + \mathcal{L}_{\text{perceptual}}(x, \hat{x}).\end{aligned} \tag{4.1}$$

Here, $sg[.]$ represents the stop-gradient operation, which prevents back-propagating gradients. The loss function comprises four terms: the reconstruction loss $\mathcal{L}_{\text{rec}} = \|x - \hat{x}\|_1$, comparing the original input x (radar maps) with its reconstruction $\hat{x}$ (reconstructed maps). The commitment loss, covered by the second and third terms, penalizes discrepancies between the encoded representations and codebook entries and optimises the codes within the codebook. The fourth term, perceptual loss, assesses high-level semantic differences between x and $\hat{x}$ [208]. The VQ-GAN is optimised training adversarially VQ-VAE, which acts as a generator, and a patch-based discriminator (i.e., spatial discriminator), using the following loss function:

$$\underset{E,G,\mathcal{Z}}{\operatorname{argmin}} \max_{D} \mathbb{E}_{x\sim p(x)} \left[\mathcal{L}_{\text{vq-vae}}(E, G, \mathcal{Z}) + \lambda \mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D)\right], \tag{4.2}$$

$$\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) = [\log D(x) + \log(1 - D(\hat{x}))] \tag{4.3}$$

$$\lambda_{\text{GAN}} = \frac{\nabla_G [\mathcal{L}_{rec}]}{\nabla_G [\mathcal{L}_{GAN}] + \delta}. \tag{4.4}$$

Here, $\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D)$ represents the discriminator loss and λ_{GAN} is the adaptive weight determined by $\nabla_G[.]$, which represents the gradient of the input concerning the final layer of decoder. $\delta = 10^{-6}$ is a scalar for numerical stability.

Autoregressive Transformer: The AT architecture aims to model the dynamics

between consecutive precipitation maps [5]. The ground truth precipitation maps are quantized into $\mathbf{z}_q = q(\mathbf{E}(x))$, producing a sequence $\mathbf{s} \in \{0, \ldots, |\mathbf{Z}| - 1\}^{h \times w}$, representing VQ-GAN codebook indices. These indices are transformed into continuous vectors by an embedder and augmented with positional embeddings to provide order information. The transformer then processes these vectors, with the head module refining the output into logits, which represent the probability of using a specific token. These logits are used to compute a cross-entropy loss that compares predicted token probabilities with the actual tokens:

$$\mathbb{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)}[-\log \prod_{i=1}^{N} p(\mathbf{s}_i | \mathbf{s}_{<i})]. \tag{4.5}$$

Given a sequence of indices $\mathbf{s}_{<i}$, the transformer is trained to predict the distribution of the consecutive indices $\mathbf{s}_i$. The AT employs a causal attention mechanism which accesses only previously seen and current tokens when predicting the next one in a sequence, enabling efficient and context-sensitive output production.

4.3.2. Physics-Informed Discriminator: PID-GAN

Following the PID-GAN [230] framework, we use physics residuals to compute a physics consistency score (η) for each prediction, indicating the likelihood of the prediction being physically consistent. These physics consistency scores are fed into a temporal discriminator [223] as additional inputs, such that it distinguishes between real and fake sequences of precipitation maps by learning from the underlying distribution of labelled points and using the additional physics supervision.

Estimating Physics Consistency Scores: Formally, we compute the physics consistency score of a prediction $\hat{x}$ for the k-th physical constraints as $\eta_k = e^{-\lambda \mathcal{R}^{(k)}(x, \hat{x})}$, where $\mathcal{R}^{(k)}$ represents the physical equation used to describe the phenomena. The larger η_k, the more prediction $\hat{x}$ obeys the k-th physical constraint. Following [230], the temporal physics-informed discriminator is trained by optimising:

$$\mathcal{L}_D(\phi) = -\frac{1}{N} \sum_{i=1}^{N} \log(D(x_i, \eta_i)) - \frac{1}{N} \sum_{i=1}^{N} \log(1 - D(\hat{x}_i, \hat{\eta}_i)). \tag{4.6}$$

Here, η_i and $\hat{\eta}_i$ represent the physics consistency score for ground truth and generated data, ensuring alignment with physical laws.

Moisture Conservation Equation: The Moisture Conservation Equation [233, 234] in NWP, which describes the relationship between atmospheric moisture content, evaporation, and precipitation (P), is given by:

$$\frac{\partial q}{\partial t} = -u \frac{\partial q}{\partial x} - v \frac{\partial q}{\partial y} - \omega \frac{\partial q}{\partial z} + ET - P \tag{4.7}$$

$$ET = 0.65 \frac{\Delta}{\Delta + \gamma} \frac{R_s}{\lambda}. \tag{4.8}$$

Here, u is the west-to-east wind component (m/s), v is the south-to-north wind component (m/s), ω is the vertical wind component (m/s), q is the specific humidity $(g\ g^{-1})$, ET is evapotranspiration rate (mm/h), Δ is the derivative with respect to temperature of the saturation vapour pressure $(Pa/°C)$, γ is the psychrometric constant (J/m^2), λ is the latent heat of vaporization (J/g), and R_s is global radiation $(Pa/°C)$. The evapotranspiration can be estimated using the Makkink equation 4.8 [235], which relies on temperature and solar radiation data, yielding relatively accurate results in cold and temperate humid climates. Measuring vertical wind speed ω poses challenges due to its low intensity and high spatial variability, the requirement for sensitive equipment, atmospheric stability, and the high costs and complexity of accurate measurements, resulting in a lack of datasets with these measurements. Omitting the term $\omega\frac{\partial q}{\partial z}$ simplifies the process but risks overlooking crucial moisture transport between atmospheric layers, potentially impacting the balance of moisture within the atmosphere's three-dimensional dynamics. To compensate for the lack of vertical wind speed data, the model shifts focus to the horizontal wind components (U and V) at different elevation levels, using the ERA5 dataset for wind measurements at 100 meters (u_{100} and v_{100}) and AWS data at 10 meters (u_{10} and v_{10}). It's important to note that atmospheric interactions, which significantly influence weather patterns, extend up to the end of the troposphere, approximately 10 km in altitude. Therefore, by focusing on wind speed within the lower 100 meters, the approach inevitably entails a degree of uncertainty, given the comprehensive atmospheric interactions occurring beyond this range. By integrating these horizontal components from both 10-meter and 100-meter altitudes, the model adapts to variations in altitude for atmospheric moisture analysis. This approach simplifies equation 4.7 to approximate moisture transport across altitudes without needing direct vertical wind speed measurements, offering a more detailed view of the atmosphere's moisture dynamics. As a result, we can define the physical constraint for the proposed PID-GAN as:

$$\mathcal{R}_q = -\frac{\partial q}{\partial t} - u_{10}\frac{\partial q}{\partial x} - v_{10}\frac{\partial q}{\partial y} - u_{100}\frac{\partial q}{\partial x} - v_{100}\frac{\partial q}{\partial y} + ET - P. \tag{4.9}$$

where the equation is computed at the pixel level.

4.4. EXPERIMENTS

Our investigation addresses two main research questions: (a) Can physical supervision improve the accuracy of precipitation nowcasting? (b) Does incorporating physical data into the model's design enhance its ability to detect extreme precipitation events? We compare the proposed model to a classic benchmark, namely Pysteps [201], a temporally consistent video prediction benchmark, TECO [236], and two benchmarks that use Extreme Value loss regularization, namely Nuwä-EVL [222] and NowcastingGPT+EVL [237]. The nowcasting task is configured with parameters set to $N = 3$ conditioning timesteps and $M = 6$ predicted maps, with each timestep representing a realistic scenario of 30 minutes. Outputs from all models are the average ensembles of five sample predictions. The PySteps model employed in this study adheres to the probabilistic configuration parameters outlined in [214].

To quantitatively assess the predictions we calculate visual fidelity metrics including Mean Squared and Absolute Errors (MSE, MAE) and Pearson Correlation Score

Table 4.1: 3-hour averaged precipitation nowcasting skill of different models (Pixel-level evaluation). Top and second-best performances are highlighted in bold and underlined, respectively. PID-GAN(-P) represents the proposed model without physical constraints, and PID-GAN(-PT) represents the proposed model without physical constraints and temporal discriminator. NGPT stands for NowcastingGPT.

	PySTEPs	TECO	Nuwä-EVL	NGPT+EVL	NGPT	PID-GAN	PID-GAN(-P)	PID-GAN(-PT)
PCC (↑)	0.219	0.149	0.202	0.253	0.241	**0.313**	<u>0.288</u>	0.250
MAE (↓)	0.798	**0.664**	0.938	0.714	0.725	<u>0.686</u>	0.706	0.692
MSE (↓)	4.210	3.335	3.592	3.298	3.293	**3.117**	<u>3.162</u>	3.271
CSI(1mm) (↑)	0.250	0.097	0.262	0.267	0.21	**0.313**	<u>0.296</u>	0.234
CSI(8mm) (↑)	0.008	0.001	0.006	<u>0.009</u>	0.005	**0.011**	0.008	0.004
FAR(1mm) (↓)	0.617	0.662	0.623	0.587	<u>0.579</u>	0.583	0.601	**0.549**
FAR(8mm) (↓)	0.592	**0.361**	<u>0.399</u>	0.502	0.513	0.529	0.499	0.435
FSS(1km) (↑)	0.375	0.163	0.394	<u>0.432</u>	0.414	**0.451**	0.430	0.428
FSS(10km) (↑)	0.467	0.211	0.456	0.493	0.463	**0.534**	<u>0.510</u>	0.481
FSS(20km) (↑)	0.522	0.248	0.498	0.534	0.508	**0.591**	<u>0.565</u>	0.521
AUC (↑)	0.454	0.378	0.516	<u>0.538</u>	0.510	**0.567**	0.532	0.520

(PCC), and nowcasting metrics, such as Critical Success Index (CSI), False Alarm Ratio (FAR) and Fractional Skill Score (FSS). Precipitation thresholds of 1 and 8 mm are set for CSI and FAR, and FSS is evaluated at spatial scales of 1, 10, and 20 km with a 1 mm threshold. Since fidelity metrics cannot capture extreme event classification, we plot a precision-recall curve of the extremes to assess the considered baselines in terms of extreme classification capabilities. Table 4.1 presents a comparison of nowcasting downstream performances for the validated models across the entire study area. The PID-GAN model's advancements are particularly notable in Mean Squared Error (MSE) and Pearson Correlation Score (PCC), indicating a strong correlation with actual precipitation events and a high degree of prediction accuracy. Additionally, the model excelled in the Fractional Skill Score (FSS) across different spatial scales and significantly outperformed all benchmarks in the Critical Success Index (CSI) at both light (1mm) and heavy (8mm) precipitation thresholds. This performance suggests an enhanced ability to detect and accurately predict a wide range of precipitation events, from light showers to severe storms.

Figure 4.2 highlights the PID-GAN model's outstanding performance in detecting extreme precipitation events within 12 Dutch catchments [214], as illustrated by its precision-recall curves. These curves not only underscore the model's superior capability in forecasting extreme events when compared to all benchmarks but also reveal PID-GAN's exceptional balance between precision and recall, evidenced by achieving the highest area under the Precision-Recall curve (AUC). This is particularly noteworthy in the context of effective nowcasting, where identifying extreme weather patterns is crucial. Additionally, the analysis reveals a significant insight: removing physical constraints from the PID-GAN model (PID-GAN(-P)) leads to a decrease of 6.17% in AUC. This drop highlights the critical role that physical constraints play in enhancing the accuracy of precipitation map predictions,

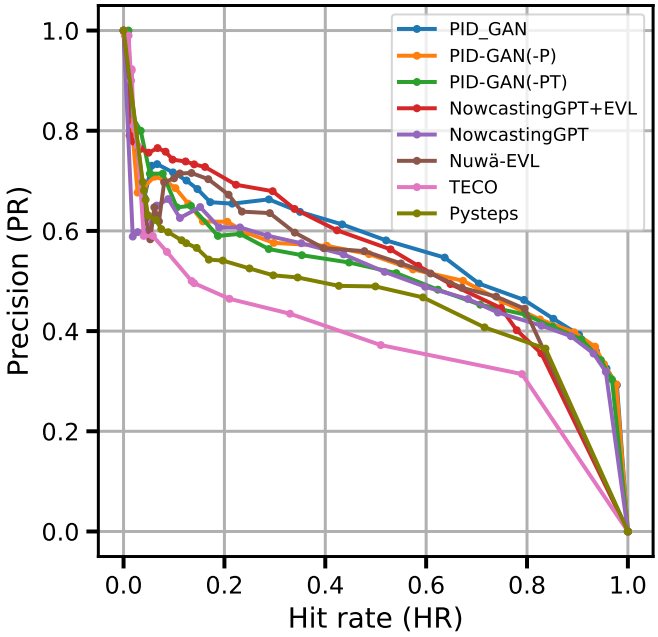


Figure 4.2: Every point from right to left represents a different precipitation threshold (0.5 to 10mm/3h) for prediction and a fixed threshold for ground truth by definition of extreme events[222].

further validating the importance of integrating physical data into the model's design for improved performance.

4.5. CONCLUSION

In conclusion, the proposed PID-GAN model has demonstrated significant effectiveness in nowcasting precipitation and accurately predicting extreme precipitation events. Addressing a notable challenge in nowcasting, the model not only outperforms existing benchmarks in terms of accuracy but also significantly enhances forecast precision. This study innovates by integrating the moisture conservation equation as a physical constraint, offering a novel approach to improving forecast accuracy. Future research will aim to further enhance the model's capabilities by incorporating additional physical constraints, such as the impact of air temperature on extreme precipitation events, and by evaluating the model's performance across diverse geographical settings.

5

MASKED GENERATIVE PRIORS IMPROVE WORLD MODELS SEQUENCE MODELLING CAPABILITIES

Won **Best Paper Award** at the ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling

Cristian Meo
Delft University of Technology, NL
c.meo@tudelft.nl

Mircea Lică
Delft University of Technology, NL

Zarif Ikram
National University of Singapore, SG

Akihiro Nakano
The University of Tokyo, JP

Vedant Shah
Mila, University of Montreal, CA

Aniket Didolkar
Mila, University of Montreal, CA

Dianbo Liu
National University of Singapore, SG

Anirudh Goyal
Mila, University of Montreal, CA

Justin Dauwels
Delft University of Technology, NL

Deep Reinforcement Learning (RL) has become the leading approach for creating artificial agents in complex environments. Model-based approaches, which are RL methods with world models that predict environment dynamics, are among the most promising directions for improving data efficiency, forming a critical step toward bridging the gap between research and real-world deployment. In particular, world models enhance sample efficiency by learning in imagination, which involves training a generative sequence model of the environment in a self-supervised manner. Recently, Masked Generative Modelling has emerged as a more efficient and superior inductive bias for modelling and generating token sequences. Building on the Efficient Stochastic Transformer-based World Models (STORM) architecture, we replace the traditional MLP prior with a Masked Generative Prior (e.g., MaskGIT Prior) and introduce GIT-STORM. We evaluate our model on two downstream tasks: reinforcement learning and video prediction. GIT-STORM demonstrates substantial performance gains in RL tasks on the Atari 100k benchmark. Moreover, we apply Categorical Transformer-based World Models to continuous action environments for the first time, addressing a significant gap in prior research. To achieve this, we employ a state mixer function that integrates latent state representations with actions, enabling our model to handle continuous control tasks. We validate this approach through qualitative and quantitative analyses on the DeepMind Control Suite, showcasing the effectiveness of Transformer-based World Models in this new domain. Our results highlight the versatility and efficacy of the MaskGIT dynamics prior, paving the way for more accurate world models and effective RL policies.

5.1. INTRODUCTION

Deep Reinforcement Learning (RL) has emerged as the premier method for developing agents capable of navigating complex environments. Deep RL algorithms have demonstrated remarkable performance across a diverse range of games, including arcade games [26, 41, 81, 86], real-time strategy games [238, 239], board games [86, 240, 241], and games with imperfect information [242]. Despite these successes, data efficiency remains a significant challenge, impeding the transition of deep RL agents from research to practical applications. Accelerating agent-environment interactions can mitigate this issue to some extent, but it is often impractical for real-world scenarios. Therefore, enhancing sample efficiency is essential to bridge this gap and enable the deployment of RL agents in real-world applications [243].

Model-based approaches [244] represent one of the most promising avenues for enhancing data efficiency in reinforcement learning. Specifically, models which learn a "world model" [245] have been shown to be effective in improving sample efficiency. This involves training a generative model of the environment in a self-supervised manner. These models can generate new trajectories by continuously predicting the next state and reward, enabling the RL algorithm to be trained indefinitely without the need for additional real-world interactions.

However, the effectiveness of RL policies trained in imagination hinges entirely on the accuracy of the learned world model. Therefore, developing architectures capable of handling visually complex and partially observable environments with

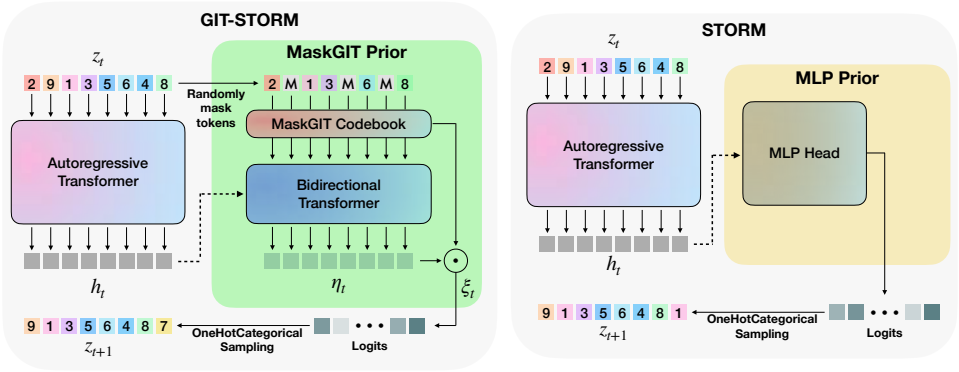


Figure 5.1: (Left) The MaskGIT prior introduced to model the dynamics of the environment. The bidirectional transformer [76] combines the hidden state given by the autoregressive transformer and the masked posterior $z_t \circ m_t$ to produce the prior corresponding to the next timestep. (Right) MLP prior originally used in STORM.

5

minimal samples is crucial. Following [245], previous methods have employed recurrent neural networks (RNN) to model the dynamics of the environment [26, 81, 84]. However, as RNNs impede parallelized computing due to their recurrent nature, some studies [39, 243, 246] have incorporated autoregressive transformer architectures [5] which have been shown to be effective across various domains, such as language [68, 73, 247, 248], images [69, 249, 250], and offline RL [251, 252]. For instance, IRIS [243] utilise discrete autoencoders [57] to map raw pixels into a smaller set of image tokens to be used as the input to the world model, achieving superhuman performance in ten different environments of the Atari 100k benchmark [253]. However, autoregressive transformers often suffer from hallucinations [254], where predicted states of the environment are unfeasible, deteriorating the agent's learning process. Additionally, their unidirectional generation process limits the ability to fully capture global contexts [255]. To address these issues, TECO [209] introduces MaskGIT [213] prior $p_\phi(z_{t+1} \mid h_t)$, using a draft-and-revise algorithm to predict the next discrete representations in the sequence in video generation task. Interestingly, STORM shows that the latent representations z_t have the biggest impact on the sequence modelling capabilities of the world model. Moreover, to the best of our knowledge, transformer-based

Table 5.1: Comparison between an MLP prior and a spatial MaskGIT prior for video dynamics using Fréchet Video Distance (FVD).

	FVD (↓)	
Method	DMLab	SSv2
TECO w/ MLP prior	153	228
TECO w/ MaskGIT prior	**48**	**199**

world models have not yet been applied to continuous action environments (e.g., DeepMind Control Suite (DMC) [253]). The primary challenge lies in the reliance on categorical latent states, which are often ill-suited for representing continuous actions. Addressing this gap is critical for extending the applicability of transformer-based world models to a broader range of tasks.

In this paper, we introduce GIT-STORM, a novel world model inspired by STORM [39], which leverages the MaskGIT prior to enhance world model sequence modelling capabilities. Building on insights from [209], we demonstrate the superior performance of the MaskGIT prior over an MLP prior in predicting video dynamics, as evidenced by results in the DMLab [256] and SSv2 [257] datasets (Table 5.1). Here we summarize the main contributions of this work:

C1: We propose GIT-STORM, a novel world model that enhances STORM [39] with a MaskGIT prior network for improved sequence modelling. Our model achieves state-of-the-art results on the Atari 100k benchmark, outperforming methods like DreamerV3 [26] and IRIS, with comprehensive ablation studies showing the impact of discrete representation quality on downstream RL tasks.

C2: We bridge the gap between transformer-based world models and continuous control tasks by using a State Mixer function that effectively combines categorical latent representations with continuous actions, enabling effective learning in continuous action spaces. Through rigorous evaluation on the DMC benchmark, we provide an in-depth analysis of the strengths and limitations of the proposed GIT-STORM model.

This paper marks a key step forward in extending transformer-based world models to more complex and diverse environments.

5.2. Related Works

5.2.1. Model-based RL: World Models

Model-based RL has been a popular paradigm of reinforcement learning. With the advent of neural networks, it has become possible to model high-dimensional state spaces and thus, use model-based RL for environments with high-dimensional observations such as RGB images. In the last few years, based on PlaNet [258], Hafner et al. proposed the Dreamer series [26, 81, 84], a class of algorithms that learn the latent dynamics of the environment using a recurrent state space model (RSSM), while learning behavioral policy in the latent space. Currently, DreamerV3 [26] has been shown to work across multiple tasks with a single configuration, setting the state-of-the-art across different benchmarks. The actor and critic in DreamerV3 learn from abstract trajectories of representations predicted by the world model.

With the advent of transformers [5] in sequence modelling and the promise of scaling performance across multiple tasks with more data, replacing the traditional RSSM backbones with transformer-based backbones has become a very active research direction. Although IRIS [243], one of the first transformer-based world model approaches, obtains impressive results, its actor-critic operates in the RGB pixel space, making it almost 14x slower than DreamerV3. In contrast, methods such as TWM [246] and STORM [39], use latent actor-critic input space. The proposed

GIT-STORM employs it as well, as we believe it is the most promising direction to overcome sample efficiency constraints. More recently, STORM updated DreamerV3 by utilizing the transformer backbone. All aforementioned transformer-based world models use an MLP head to model a dynamics prior which is used to predict the discrete representation of the following timestep. In contrast, introduced by TECO [209], we employ a MaskGIT [213] prior head, which enhances the sequence modelling capabilities of the world model. Table 5.2 compares various design aspects of different world models. Furthermore, besides STORM, all the mentioned transformer-based world models concatenate the discrete action to the extracted categorical latent representations. As a result, none of these methods is able to handle continuous actions. In contrast, combining latent representations and actions with a state mixer, we successfully train STORM and GIT-STORM on a challenging continuous action environment (i.e., DMC).

5.2.2. MASKED MODELLING FOR VISUAL REPRESENTATIONS AND GENERATION

Inspired by the Cloze task [259], BERT [76] proposed a masked language model (MLM) pre-training objective that led to several state-of-the-art results on a wide class of natural language tasks. Following the success of BERT, Masked Autoencoders (MAEs) [249] learn to reconstruct images with masked patches during the pre-training stage. The learned representations are then used for downstream tasks. [260] similarly, improves upon a BERT-like masking objective for its non-autoregressive generation algorithm.

The most relevant to our work is MaskGIT [213], a non-autoregressive decoding approach that consists of a bidirectional transformer model, trained by learning to predict randomly masked visual tokens. By leveraging a bidirectional transformer [76], it can better capture the global context across tokens during the sampling process. Furthermore, training on masked token prediction enables efficient, high-quality sampling at a significantly lower cost than autoregressive models. MaskGIT achieves state-of-the art performance on ImageNet dataset and achieves a 64× speed-up on autoregressive decoding. The MaskGIT architecture has been applied to various tasks, such as video generation [64, 209, 261] and multimodal generation [70]. For example, [209] proposes TECO, a latent dynamics video prediction model that uses MaskGIT to model the prior for predicting the next timestep discrete representations, enhancing the sequence modelling of a backbone autoregressive transformer. Inspired by TECO, we adopt the use of MaskGIT prior for the world model, enhancing the sequence modelling capabilities, crucial for enabling and improving the agent policy learning behavior.

Further discussion of related works can be found in Appendix 5.7.

5.3. METHOD

Following DreamerV3 [26] and STORM [39], we define our framework as a partially observable Markov decision process (POMDP) with discrete timesteps, $t \in \mathbb{N}$, scalar rewards, $r_t \in \mathbb{R}$, image observations, $o_t \in \mathbb{R}^{h \times w \times c}$, and discrete actions.

Table 5.2: AC stands for Actor Critic, OneHot for OneHotCategorical.

Module	DreamerV3 [26]	IRIS [243]	TWM [246]	STORM [39]	GIT-STORM (ours)
Latent space	[OneHot, Hidden]	VQ Codes	OneHot	OneHot	OneHot
Dynamics Model	RSSM	Transformer	TransformerXL	Transformer	Transformer
Dynamics Prior	MLP	MLP	MLP	MLP	MaskGIT
AC Input Space	[Latent, Hidden]	RGB	Latent	[Latent, Hidden]	[Latent, Hidden]
Experience Sampling	Uniform	Uniform	Balanced	Uniform	Uniform

$a_t \in \{1, \dots, m_a\}$. These actions are governed by a policy, $a_t \sim \pi(a_t \mid o_{1:t}, a_{1:t-1})$, where $o_{1:t}$ and $a_{1:t-1}$ represent the previous observations and actions up to timesteps t and $t-1$, respectively. The termination of each episode is represented by a Boolean variable, $c_t \in \{0, 1\}$. The goal is to learn an optimal policy, π, that maximises the expected total discounted rewards, $\mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\right]$, where $\gamma \in [0, 1]$ serves as the discount factor. The learning process involves two parallel iterative phases: learning the observation and dynamics modules (World Model) and optimising the policy (Agent).

In this section, we first provide an overview of the dynamics module of GIT-STORM. Then, we describe our dynamics prior head of the dynamics module, inspired by MaskGIT [213] (Figure 5.1). Finally, we explain the imagination phase using GIT-STORM, focusing on the differences between STORM and GIT-STORM. We follow STORM for the observation module and DreamerV3 for the policy definition, which are described in Appendix 5.7 and 5.7, respectively.

5.3.1. Overview: Dynamics Module

The dynamics module receives representations from the observation module and learns to predict future representations, rewards, and terminations to enable planning without the usage of the observation module (imagination). We implement the dynamics module as a Transformer State-Space Model (TSSM). Given latent representations from the observation module, z_t, and actions, a_t, the dynamics module predicts hidden states, h_t, rewards, $\hat{r}_t$, and episode termination flags, $\hat{c}_t \in \{0, 1\}$ as follows,

$$
\begin{aligned}
\zeta_t &= g_\theta(z_t, a_t) && \text{(State Mixer)} \\
h_t &= f_\theta(\zeta_{1:t}) && \text{(Autoregressive Transformer)} \\
z_{t+1} &\sim p_\phi(z_{t+1} \mid h_t) && \text{(Dynamics Prior Head)} \\
\hat{r}_t &\sim p_\phi(\hat{r}_t \mid h_t) && \text{(Reward Head)} \\
\hat{c}_t &\sim p_\phi(\hat{c}_t \mid h_t) && \text{(Termination Head)}
\end{aligned}
\tag{5.1}
$$

The world model is optimised to minimise the objective,

$$\mathcal{L}(\phi) = \frac{1}{BT} \sum_{n=1}^{B} \sum_{t=1}^{T} \left[\mathcal{L}_{\text{rew}}(\phi) + \mathcal{L}_{\text{term}}(\phi) + \beta_1 \mathcal{L}_{\text{dyn}}(\phi) + \beta_2 \mathcal{L}_{\text{rep}}(\phi) \right] \tag{5.2}$$

where β_1, β_2 are loss coefficients and $\mathcal{L}_{\text{rew}}(\phi), \mathcal{L}_{\text{term}}(\phi), \mathcal{L}_{\text{rep}}(\phi), \mathcal{L}_{\text{dyn}}(\phi)$ are reward, termination, representation, and dynamics losses, respectively. We use the symlog two-hot loss described in [26] as the reward loss. The termination loss is calculated as cross-entropy loss, $c_t \log \hat{c}_t + (1 - c_t) \log(1 - \hat{c}_t)$. In the following section, we define the dynamics prior in Eq. 5.1, as well as representation loss, $\mathcal{L}_{\text{rep}}$, and dynamics loss, $\mathcal{L}_{\text{dyn}}$.

5.3.2. DYNAMICS PRIOR HEAD: MASKGIT PRIOR

Given the expressive power of MaskGIT [213], we propose enhancing the dynamics module in the world model by replacing the current MLP prior with a MaskGIT prior, as shown in Figure 5.1. Given the posterior, z_t, and a randomly generated mask, $m \in \{0, 1\}^N$ with $M = \lceil \gamma N \rceil$ masked values where $\gamma = \cos\left(\frac{\pi}{2} t\right)$, the MaskGIT prior $p_\phi(z_{t+1} \mid h_t)$ is defined as follows.

First, the hidden states, h_t, are concatenated with the masked latent representations, $z_t \circ m_t$, where $\circ$ indicates element-wise multiplication. Despite h_t being indexed by t, it represents the output of the f_θ and thus encapsulates information about the subsequent timestep. Consequently, the concatenation of z_t and h_t integrates information from both the current and the next timestep, respectively. A bidirectional transformer is then used to learn the relationships between these two consecutive representations, producing a summary representation, ξ_t. Finally, logits are computed as the dot product (denoted as $\odot$ in Figure 5.1) between the MaskGIT embeddings, which represent the masked tokens, and ξ_t. This dot product is also known as weight tying strategy, first formalized in [262] and then used in the original MaskGIT [213] and GPT-2 [247] models as well because of its regularization effects that help preventing overfitting [262]. Indeed, this weight tying strategy (i.e., dot product) can be interpreted as a similarity distance between the embeddings and ξ_t. Indeed, from a geometric perspective, both cosine similarity and the dot product serve as similarity metrics, with cosine similarity focusing on the angle between two vectors, while the dot product accounts for both the angle and the magnitude of the vectors. Therefore, by optimising the MaskGIT prior, this dot product aligns the embeddings with ξ_t, thereby facilitating and improving the computation of logits. In contrast, when using the MLP prior, the logits are generated as the output of an MLP that only takes h_t as input. This approach requires the model to learn the logits space and their underlying meaning without any inductive bias, making the learning process more challenging.

During training, we follow the KL divergence loss of DreamerV3 [26], which consists of two KL divergence losses which differ in the stop-gradient operator, $\text{sg}(\cdot)$, and loss scale. We account for the mask tokens in the posterior and define $\mathcal{L}_{\text{dyn}}$ and

$\mathcal{L}_{\text{rep}}$ as,

$$\mathcal{L}_{\text{dyn}}(\phi) \doteq \max\big(1, \mathrm{KL}\big[\,\mathrm{sg}(q_\phi(z_t \mid x_t)) \circ m_t \,\big\|\, p_\phi(z_t \mid h_{t-1})\,\big]\big) \tag{5.3}$$

$$\mathcal{L}_{\text{rep}}(\phi) \doteq \max\big(1, \mathrm{KL}\big[\,q_\phi(z_t \mid x_t) \circ m_t \,\big\|\, \mathrm{sg}(p_\phi(z_t \mid h_{t-1}))\big]\big) \tag{5.4}$$

where m_t is multiplied element-wise with the posterior, eliminating the masked tokens from the loss.

Sampling. During inference, since MaskGIT has been trained to model both unconditional and conditional probabilities, we can sample any subset of tokens per sampling iteration. Following [209], we adopt the Draft-and-Revise decoding scheme introduced by [255] to predict the next latent state (Algorithm 1 and 2). During the draft phase, we initialize a partition $\mathbf{\Pi}$ which contains T_{draft} disjointed mask vectors $\mathbf{m}$ of size (latent dim $\div\, T_{\text{draft}}$), which together mask the whole latent representation. Iterating through all mask vectors in $\mathbf{\Pi}$, the resulting masked representations are concatenated with the hidden states h_t from Eq. 5.1 and fed to the MaskGIT prior head that computes the logits of the tokens correspondent to h_t and $\mathbf{m}^i$. Such logits are then used to sample the new tokens that replace the positions masked by $\mathbf{m}^i$. During the revise phase, the whole procedure is repeated Γ times. As a result, when sampling the new tokens, the whole representation is taken into account, resulting in a more consistent and meaningful sampled state.

5.3.3. State Mixer for Continuous Action Environments

When using a TSSM as the dynamics module, the conventional approach has been to concatenate discrete actions with categorical latent representations and feed this sequence into the autoregressive transformer. However, this method is ineffective for continuous actions, as one-hot categorical representations or VQ-codes [57] are poorly suited for representing continuous values. To overcome this limitation, we repurpose the state mixer function $g_\theta(\cdot)$ introduced in STORM, which combines the latent representation and the action into a unified mixed representation ζ_t. This approach allows for the integration of both continuous and discrete actions with latent representations, enabling the application of TSSMs to environments that require continuous action spaces.

Algorithm 1

Require: Partition sampling distributions p_{draft} and p_{revise}, the number of revision iterations Γ, hidden states h_t, model θ

/* draft phase */

$\mathbf{z}^{\text{empty}} \leftarrow (\texttt{[MASK]}, \cdots, \texttt{[MASK]})^N$

$\mathbf{\Pi} \sim p_{\text{draft}}(\mathbf{\Pi}; T_{\text{draft}})$

/* generate a draft prior map */

$\mathbf{z}^0 \leftarrow \text{MASKGIT HEAD}(\mathbf{z}^{\text{empty}}, \mathbf{\Pi}, h_t; \theta)$

/* revision phase */

for $\gamma = 1, \cdots, \Gamma$ **do**

  $\mathbf{\Pi} \sim p_{\text{revise}}(\mathbf{\Pi}; T_{\text{revise}})$

  $\mathbf{z}^{\gamma} \leftarrow \text{MASKGIT HEAD}(\mathbf{z}^{\gamma-1}, \mathbf{\Pi}, h_t; \theta)$

end for

$\mathbf{z}_{t+1} \leftarrow \mathbf{z}^{\Gamma}$

return $\mathbf{z}_{t+1}$

Algorithm 2

Require: Generated latents $\mathbf{z}$, hidden states h_t, partition $\mathbf{\Pi} = (\mathbf{m}^1, \ldots, \mathbf{m}^T)$, model θ

▷ Update the codes

for $i = 1$ to T **do**

  MaskGIT_codes $\leftarrow$ MaskGIT_Codebook($\mathbf{z} \circ \mathbf{m}^i$)

  $\xi \leftarrow$ BidirectionalTransformer(MaskGIT_codes, h_t)

  logits $\leftarrow \xi \odot$ MaskGIT_embeddings

  $\hat{\mathbf{z}} \sim$ Categorical(logits)

  $\mathbf{z} \leftarrow (1 - \mathbf{m}^i) \circ \mathbf{z} + \mathbf{m}^i \circ \hat{\mathbf{z}}$

end for

return $\mathbf{z}$

5.3.4. IMAGINATION PHASE

Instead of training the policy by interacting with the environment, model-based approaches use the learned representation of the environment and plan in imagination [258]. This approach allows sample-efficient training of the policy by propagating value gradients through the latent dynamics. The interdependence between the dynamics generated by the world model and agent's policy makes the quality of the imagination phase crucial for learning a meaningful policy. The imagination phase is composed of two phases, conditioning phase and the imagination one. During the conditioning phase, the discrete representations z_t are encoded and fed to the autoregressive transformer. The conditioning phase gives context for the imagination one, using the cached keys and values [60] computed during the conditioning steps.

Differently from STORM, which uses a MLP prior to compute the next timestep representations, we employ MaskGIT to accurately model the dynamics of the

environment. By improving the quality of the predicted trajectories, the agent is able to learn a superior policy.

5.4. EXPERIMENTS

In this section, we analyse the performance of GIT-STORM and its potential limitations by exploring the following questions: (a) How does the MaskGIT Prior affect TSSMs learning behaviour and performances on related downstream tasks (e.g., Model-based RL and Video Prediction tasks)? (b) Can Transformer-based world models learn to solve tasks on continuous action environments when using state mixer functions?

5.4.1. EXPERIMENTAL SETUP

To evaluate and analyse the proposed method, we consider both discrete and continuous actions environments, namely Atari 100k benchmark [253] and DeepMind Control Suite [263] respectively. On both environments, we conduct both RL and video prediction tasks.

Benchmark and baselines. Atari 100k benchmark consists of 26 different video games with discrete action space. The constraint of 100k interactions corresponds to a total of 400k frames used for training, as frame skipping is set to 4. For RL task on Atari 100k benchmark, we compare GIT-STORM against one model-free method, SimPLe [253], one RSSM, DreamerV3 [26], and three TSSM models (i.e., IRIS [243], TWM [246], and STORM [39]). DMC benchmark consists of 18 control tasks with continuous action space. We restrict the models to be trained with only 500k interactions (1M frames) by setting frame skipping to 2. For RL task on DMC benchmark, we compare our model against SAC [264], CURL [43], DrQ-v2 [265], PPO [266], DreamerV3 [26], and STORM [39]. We trained GIT-STORM on 5 different seeds. For video prediction tasks, we compare GIT-STORM with STORM only to understand how the MaskGIT Prior affects the visual quality of predicted frames and its influence on the policy training.

Extended details of the baselines for both benchmarks can be found in Appendix 5.7.

Evaluation metrics. Proper evaluation of RL algorithms is known to be difficult due to both the stochasticity and computational requirements of the environments [267]. To provide an accurate evaluation of the models, we consider a series of metrics to assess the performances of the considered baselines on across the selected experiments. We report human normalized mean and median as evaluation metrics, aligning with prior literature. We also report interquartile Mean (IQM), Optimality Gap, Performance Profiles (scores distributions), and Probability of Improvement (PI), which provide a statistically grounded perspective on the model evaluation [267].

For video prediction task, we report two metrics: Fréchet Video Distance (FVD) [188] to evaluate visual quality of the predicted frames, and perplexity [268] measure of

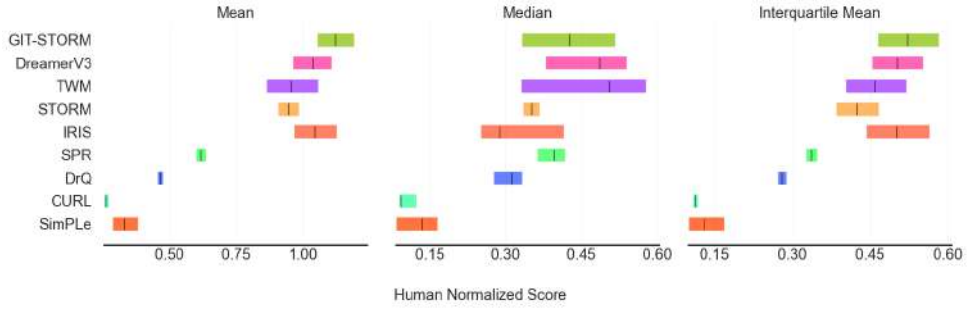


Figure 5.2: (Left) Human normalized mean, across the Atari 100k benchmark. GIT-STORM outperforms all other baselines. (Middle) Human normalized median. TWM achieves the highest median value of 51%. (Right) IQM. GIT-STORM outperforms all other baselines.

the predicted tokens to evaluate the token utilization by the dynamics prior head. We use the trained agent to collect ground truth episodes and use the world model to predict the frames. We report the FVD over 256 videos which are conditioned on the first 8 frames to predict 48 frames.

A full description of these metrics can be found in Appendix 5.7.

5.4.2. RESULTS ON DISCRETE ACTION ENVIRONMENTS: ATARI 100K

RL task. Figure 5.2 summarizes the human normalized mean and median, and IQM score. The full results on individual environments can be found in the Appendix due to space limitations (Table 5.5). We can see that while TWM and DreamerV3 present a higher human median than GIT-STORM (TWM: 51%, DreamerV3: 49% $\rightarrow$ GIT-STORM: 42.6%), GIT-STORM dominates in terms of human mean (TWM: 96%, DreamerV3: 104% $\rightarrow$ GIT-STORM: 112.6%). In terms of IQM, a more robust and statistically meaningful metric, GIT-STORM significantly outperforms the related baselines (DreamerV3: 0.501, IRIS: 0.502 $\rightarrow$ GIT-STORM: 0.522).

Figure 5.3 (Left) compares PI against the baselines. Noticeably, GIT-STORM presents $PI > 0.5$ for all baselines, which indicates that, from a probabilistic perspective GIT-STORM would outperform each baseline on a random task Y from Atari 100k with a probability greater than 0.5. Figure 5.8 illustrates the Optimality Gap, while Figure 5.9 presents the fraction of runs with score $> \tau$ for different human normalized scores; both confirm the trends observed so far. Moreover, a closer look to Table 5.5 reveals that GIT-STORM presents an optimality gap of 0.500, marginally beating DreamerV3, which reports 0.503 and significantly outperforming all other baselines.

Video Prediction task. Table 5.3 shows video prediction results on selected Atari 100k environments. The table shows that GIT-STORM presents, on average, lower FVD and higher perplexity than STORM (e.g., in Freeway, STORM: 105.45, 33.15 $\rightarrow$

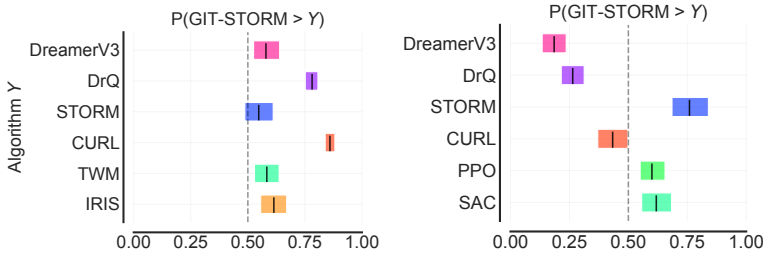


Figure 5.3: Probability of Improvement of the mentioned baselines and GIT-STORM in the Atari 100k benchmark (Left) and DMC benchmark (Right). The results represent how likely it is for GIT-STORM to outperform other baselines.

GIT-STORM: 80.33, 67.92, respectively). Figure 5.5 shows several video prediction results on each environment. For example, on Boxing, we can see that GIT-STORM is able to predict more accurately into the future. The differences in the other two games are smaller, as the player in each game has a much smaller dimension. We think GIT-STORM achieves higher perplexity because the learned agent can collect more diverse episodes.

5.4.3. Results on Continuous Action Environments: DeepMind Control Suite

RL task. Figure 5.4 summarizes the human normalized mean and median, and IQM score. The full results on individual environments can be found in the Appendix (Table 5.6). Although DreamerV3 outperforms all other models on average, Table 5.6 shows that GIT-STORM presents state-of-the-art scores on two environments, Walker Stand and Quadruped Run. Compared to STORM, GIT-STORM consistently and significantly outperforms across the whole benchmark in terms of human median and mean (STORM: 31.50, 214.50 → GIT-STORM: 475.12, 442.10, respectively). For PI, GIT-STORM achieves $PI > 0.5$ than STORM, PPO, and SAC (e.g., GIT-STORM: 0.75, 0.60 and 0.63, over STORM, PPO and SAC, respectively) (Figure 5.3 (Right)).

Table 5.3: FVD and perplexity comparisons of STORM and GIT-STORM on selected Atari 100k environments.

Game	FVD (↓)		Perplexity (↑)	
	STORM	GIT-STORM	STORM	GIT-STORM
Boxing	**1458.32**	1580.32	49.24	**54.95**
Hero	381.16	**354.16**	10.55	**30.25**
Freeway	105.45	**80.33**	33.15	**67.92**

Table 5.4: FVD and perplexity comparisons of STORM and GIT-STORM on selected DMC environments.

Task	FVD (↓)		Perplexity (↑)	
	STORM	GIT-STORM	STORM	GIT-STORM
Cartpole Balance Sparse	2924.81	**1892.44**	1.00	**3.76**
Hopper Hop	4024.11	**3458.19**	3.39	**22.59**
Quadruped Run	3560.33	**1000.91**	1.00	**2.61**

Video Prediction task. Table 5.4 shows video prediction results on selected DMC environments. The table shows that our model achieves lower FVD and higher perplexity than STORM for all environments. The video prediction results in Figure 5.6 show that although both models fail to capture the dynamics accurately, GIT-STORM generates marginally better predictions, leading to higher perplexity as well.

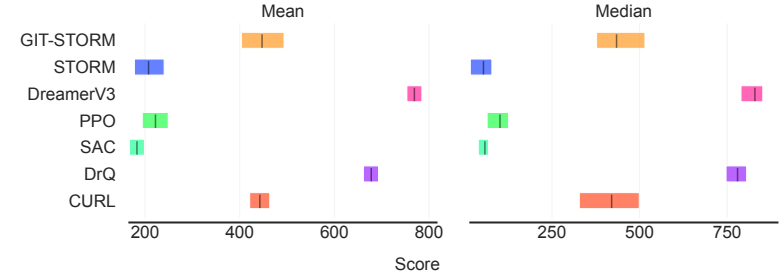


Figure 5.4: Comparison of human normalized mean (left) and median (right) on DMC benchmark.

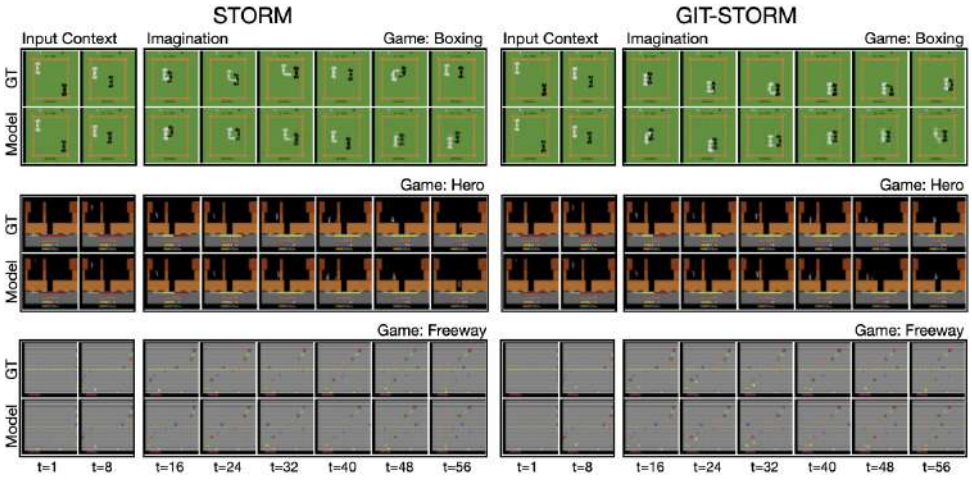


Figure 5.5: Generated trajectories of STORM and GIT-STORM on selected Atari 100k environments. The model uses the first 8 frames as context and then generates the following 48 frames.

5.5. DISCUSSION

The proposed GIT-STORM uses a Masked Generative Prior (MaskGIT) to enhance the world model sequence modelling capabilities. Indeed, as discussed in the introduction, high quality and accurate representations are essential to guarantee and enhance

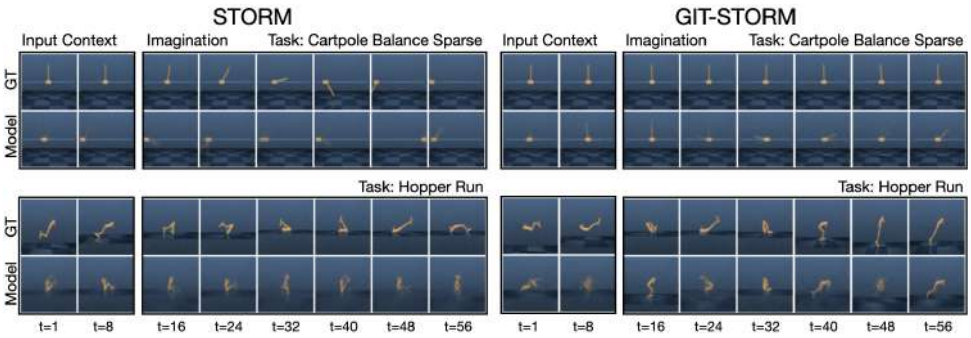


Figure 5.6: Generated trajectories of STORM and GIT-STORM on selected DMC environments. The model uses the first 8 frames as context and then generates the following 48 frames.

agent policy learning in imagination. Remarkably, the proposed GIT-STORM is the only world model, among the ones that use uniform sampling and latent actor critic input space, that is able to achieve non-zero reward on the Freeway environment (e.g., DreamerV3: 0, STORM: 0 $\rightarrow$ GIT-STORM: 13). Indeed, both STORM and IRIS resorted in ad-hoc solutions to get positive rewards, such as changing the sampling temperature [243] and using demonstration trajectories [39]. Such result, together with the quantitative results on the Atari 100k and DMC benchmarks, clearly answer question (a) - the presented MaskGIT prior improves the policy learning behaviour and performance on downstream tasks (e.g., Model-based RL and Video Prediction) of TSSMs. Moreover, the FVD and perplexity comparisons in Table 5.3 and Table 5.4 suggest that GIT-STORM has better predictive capabilities, learns a better dynamics module, and presents more accurate imagined trajectories (Figure 5.5, Figure 5.6). Similarly to image synthesis [213] and video prediction [64] tasks, we show how using masked generative modelling is a better inductive bias to model the prior dynamics of discrete representations and improve the downstream usefulness of world models on RL tasks. Furthermore, the MaskGIT Prior can be used in any sequence modelling backbone that uses categorical latent representations (e.g., VideoGPT [60], IRIS [243]), positioning itself as a very versatile approach. In this work we do not apply a MaskGIT prior on top of IRIS only because of computational and time constraints - IRIS requires 168h of training on a V100 GPU for a single run [39] .

Noticeably, the quantitative results on DMC benchmarks answer question (b) - It is possible to train TSSMs when using a mixer function to combine categorical representations and continuous actions. Indeed, both STORM and GIT-STORM are able to learn meaningful policies within the DMC benchmark. Remarkably, GIT-STORM outperforms STORM with an substantial margin, while using exactly the same policy learning algorithm. Interestingly, Figure 5.15 presents an ablation of the used state mixer function, revealing that the overall learning behaviourhighly depends on the used inductive bias. Surprisingly, the simplest one (e.g., concatenation of z_t and a_t) is the only one that works meaningfully. We leave the exploration

of better inductive biases (e.g., imposing specific information bottlenecks [269]) to improve the state mixer function as future work.

Limitations and Future Work. The current implementation has been validated on environments that do not require extensive training steps (e.g., ProcGen [270], Minecraft [271]) to be trained. We keep as a future work the validation of GIT-STORM on ProcGen and Minecraft environments. As suggested by [209], using a MaskGIT prior could benefit the world model learning behaviourin a visually challenging environment like Minecraft. From a technical point of view, one of the main limitations of the proposed world model is that we use only one iteration for the Draft-and-Revise decoding scheme [255]. Indeed, while using one iteration speeds up training and evaluation, we do not fully exploit the advantages of this decoding scheme. As a result, in environments like Pong or Breakout, which present small objects (e.g., white or red balls, respectively), using a masked generative approach can lead to filtering such objects out, degrading the downstream performances in these environments. The main reason is that the presented decoding scheme scales exponentially with the number of iterations. We leave as future work the definition of a decoding scheme that scales more efficiently with the number of iterations.

5.6. Conclusion

The motivation for this work stems from the need to improve the quality and accuracy of world models representations in order to enhance agent policy learning in challenging environments. Inspired by [209], we conducted experiments using the TECO framework on video prediction tasks with DMLab and SSv2 [257] datasets. Replacing an MLP prior with a MaskGIT [213] prior significantly improved the sequence modelling capabilities and the related performance on the video prediction downstream task. Building upon these insights, we proposed GIT-STORM, which employs a MaskGIT Prior to enhance the sequence modelling capabilities of world models, crucial to improve the policy learning behavior [243]. Moreover, through the use of a state mixer function, we successfully combined categorical latent representations with continuous actions, and learned meaningful policies on the related environments. We validated the proposed approach on the Atari 100k and the DMC benchmarks. Our quantitative analysis showed that GIT-STORM on average outperforms all baselines in the Atari 100k benchmark while outperforming STORM with a significant margin on the DMC benchmark. Although our approach does not beat the state-of-the-art in the DMC benchmark, the presented quantitative and qualitative evaluations led to the conclusion that masked generative priors (e.g., MaskGIT Prior) improve world models sequence modelling capabilities and the related downstream usefulness.

5.7. APPENDIX

GIT-STORM FRAMEWORK

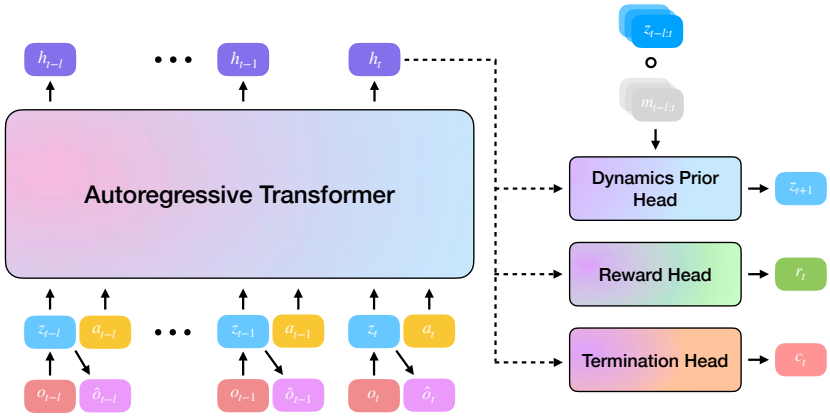


Figure 5.7: Similar to STORM [39], GIT-STORM performs sequence modelling using an autoregressive transformer, which predicts future stochastic latents, z_t, reward, r_t and termination, c_t. In contrast with STORM, GIT-STORM uses a Masked Generative Prior to model the dynamics of the environment.

Similar to previous TSSM-based world models [36, 39, 243], GIT-STORM consists of a world model with two modules, VAE-based observation module and autoregressive dynamics module, and a policy trained in the latent space. Figure 5.7 describes the world model architecture of GIT-STORM. In the following sections, we provide details of the observation module and policy.

OBSERVATION MODULE

Following STORM, the observation module is a variational autoencoder (VAE) [14], which encodes observations, o_t, into stochastic latent representations, z_t, and decodes back the latents to the image space, $\hat{o}_t$:

$$\text{Observation encoder: } z_t \sim q_\phi(z_t \mid o_t) \tag{5.5}$$

$$\text{Observation decoder: } \hat{o}_t = p_\phi(z_t) \tag{5.6}$$

The observations are encoded using a convolutional neural network (CNN) encoder [87] which outputs the logits used to sample from a categorical distribution. The distribution head applies an unimix function over the computed logits to prevent the probability of selecting any category from being zero [272]. Since the sampled latents lack gradients, we use the straight-through gradients trick [88] to preserve them. The decoder, modeled using a CNN, reconstructs the observation from the

latents, z_t. While the encoder is updated using gradients coming from both observation and dynamics modules, the decoder is optimised using only the Mean Squared Error (MSE) between input and reconstructed frames:

$$\mathcal{L}_{\text{Observation Model}} = \text{MSE}(o_t, \hat{o}_t) \tag{5.7}$$

POLICY LEARNING

Following the model-based RL research landscape [DreamerV3; 26] we cast the agent policy learning framework using the actor-critic approach [273]. The agent actor-critic is trained purely from agent state trajectories $s_t = [z_t, h_t]$ generated by the world model. The actor aims to learn a policy that maximises the predicted sum of rewards and the critic aims to predict the distribution of discounted sum of rewards by the current actor:

$$\text{Actor: } a_t \sim \pi_\theta(a_t|s_t) \ , \ \text{Critic: } V_\psi(s_t) \approx \mathbb{E}_{\pi_\theta, p_\phi}\left[\sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}\right], \tag{5.8}$$

where γ is a discount factor.

We follow the setup of STORM [39] and DreamerV3 [26] to train the agent. First, a random trajectory is sampled from the replay buffer to compute the initial state of the agent. Then, using the sampled trajectory as context, the world model and actor generate a trajectory of imagined model states, $s_{1:L}$, actions, $a_{1:L}$, rewards, $\hat{r}_{1:L}$, and termination flags, $\hat{c}_{1:L}$, where L is the imagination horizon. To estimate returns that consider rewards beyond the prediction horizon, we compute bootstrapped λ-returns [26, 80] defined recursively as follows:

$$G_l^\lambda = \hat{r}_l + \gamma \hat{c}_l \left[(1-\lambda) V_\psi(s_{l+1}) + \lambda V_{l+1}^\lambda\right] \ , \ G_L^\lambda = V_\psi(s_L) \tag{5.9}$$

To stabilise training and prevent the model from overfitting, we regularize the critic towards predicting the exponential moving average (EMA) of its own parameters. The EMA of the critic is updated as,

$$\psi_{l+1}^{\text{EMA}} = \sigma \psi_l^{\text{EMA}} + (1-\sigma)\psi_l, \tag{5.10}$$

where σ is the decay rate. As a result, the critic learns to predict the distribution of the return estimates using the following maximum likelihood loss:

$$\mathcal{L}_\psi = \frac{1}{BL} \sum_{n=1}^{B} \sum_{l=1}^{L} \left[(V_\psi(s_l) - \text{sg}(G_l^\lambda))^2 + (V_\psi(s_l) - \text{sg}(V_{\psi^{\text{EMA}}}(s_l)))^2\right], \tag{5.11}$$

The actor learns to choose actions that maximise return while enhancing exploration using an entropy regularizer [26, 274]. Reinforce estimator [275] is used for actions,

resulting in the surrogate loss function:

$$\mathcal{L}_\theta = \frac{1}{BL}\sum_{n=1}^{B}\sum_{l=1}^{L}\left[-\mathrm{sg}\left(\frac{G_l^\lambda - V_\psi(s_l)}{\max(1,S)}\right)\ln \pi_\theta(a_l|s_l) - \eta H(\pi_\theta(a_l|s_l))\right], \tag{5.12}$$

where $\mathrm{sg}(\cdot), H(\cdot)$ are stop gradient operator and entropy, respectively, and η is a hyperparameter coefficient of the entropy loss. When training the actor, the rewards are computed between the range from the 5^{th} to the 95^{th} percentile and smoothed out by using an EMA to be robust to outliers. Therefore, the normalization ratio S is,

$$S = \mathrm{EMA}(\mathrm{percentile}(G_l^\lambda, 95) - \mathrm{percentile}(G_l^\lambda, 5)). \tag{5.13}$$

EXTENDED RELATED WORKS: VIDEO PREDICTION MODELLING

Video prediction, a fundamental task in computer vision, aims to generate or predict sequences of future frames based on conditioning past frames. The downstream tasks of video prediction modelling span a wide range of domains, showcasing its significance in different fields, such as autonomous driving [58], robot navigation [276] controllable animation [277], weather forecasting [59, 237], and model based reinforcement learning [26, 39, 81, 84, 243, 258]. Video prediction modelling is known for its sample inefficiency, which poses significant challenges in learning accurate and reliable models in a feasible time [278]. To address this, recent advancements have introduced spatio-temporal state space models, which typically consist of a feature extraction component coupled with a dynamics prediction module. These models aim to understand and predict the evolution of video frames by capturing both spatial and temporal relationships. Notable examples include NUWÄ [206] and VideoGPT [60] which respectively use 2D and 3D convolutional layers to extract the latent representations and an autoregressive transformer to perform sequence modelling in the latent space. Moreover, TECO [209] introduces the use of MaskGIT [213] prior to improve the accuracy of the predicted discrete latents and uses a 1D convolution to enhance temporal consistency. Furthermore, VideoPoet [279], which is able to handle multiple modalities and perform a variety of tasks besides video prediction.

FULL RESULTS ON RL TASK

In this section we report and present the full evaluation and comparison on the two RL benchmark environments, Atari 100k [253] and DMC [263]. Table 5.5 and Table 5.6 are the results on Atari 100k and DMC, respectively.

TRAINING CURVES

In this section, we provide the training curves of GIT-STORM for both Atari 100k and DMC benchmark.

Table 5.5: We report mean scores as well as aggregated human normalized mean and median, Interquantile Mean (IQM), and Optimality Gap. Following the conventions of [81], scores that are the highest or within 5% of the highest score are highlighted in bold.

Game	Rand	Hum	SimPLe reported	TWM reported	IRIS reported	DreamerV3 reproduced	STORM reproduced	GIT-STORM ours
Alien	228	7128	617	675	420	804	**1364**	1145
Amidar	6	1720	74	122	143	122	**239**	181
Assault	222	742	527	683	**1524**	642	707	967
Asterix	210	8503	1128	1116	854	**1190**	865	811
Bank Heist	14	753	34	467	53	**752**	375	503
Battle Zone	2360	37188	4031	5068	**13074**	11600	10780	9470
Boxing	0	12	8	**78**	70	71	**80**	**81**
Breakout	2	30	16	20	**84**	24	12	12
Chopper Command	811	7388	979	1697	1565	680	2293	**2048**
Crazy Climber	10780	35829	62584	71820	59234	**86000**	54707	55237
Demon Attack	152	1971	208	350	**2034**	203	229	223
Freeway	0	30	17	24	**31**	0	0	13
Frostbite	65	4335	237	**1476**	259	1124	646	582
Gopher	258	2413	597	1675	2236	4358	2631	**8562**
Hero	1027	30826	2657	7254	7037	12070	11044	**13351**
Jamesbond	29	303	101	362	463	290	**552**	471
Kangaroo	52	3035	51	1240	838	**4080**	1716	1601
Krull	1598	2666	2204	6349	6616	**7326**	6869	**7011**
Kung Fu Master	256	22736	14862	**24555**	21760	19100	20144	**24689**
Ms Pacman	307	6952	1480	1588	999	1370	**2673**	1877
Pong	-21	15	13	**19**	15	**19**	8	6
Private Eye	25	69571	35	87	100	140	**2734**	2225
Qbert	164	13455	1289	3331	746	1875	2986	**3924**
Road Runner	12	7845	5641	9109	9615	14613	12477	**17449**
Seaquest	68	42055	683	**774**	661	571	525	459
Up N Down	533	11693	3350	**15982**	3546	7274	7985	10098
Human Mean (↑)	0%	100%	33%	96%	105%	104%	94.7%	**112.6%**
Human Median (↑)	0%	100%	13%	**51%**	29%	49%	35.7%	42.6%
IQM (↑)	0.00	1.00	0.130	0.459	0.501	0.502	0.426	**0.522**
Optimality Gap (↓)	1.00	0.00	0.729	0.513	0.512	0.503	0.528	**0.500**

ABLATION STUDY

GIT-STORM ABLATIONS

In this section, we analyse the contributions of the two primary components that define GIT-STORM:

- **MaskGIT Head**: We compare the performance of the MaskGIT head against a standard MLP head to assess its role in improving downstream results.
- **Logits Computation via Dot Product**: We evaluate the impact of computing logits as the dot product between ξ_t and the MaskGIT embeddings, comparing this approach to the alternative of using an MLP head that takes ξ_t as input and directly outputs logits.

These components are hypothesized to be critical for understanding the capabilities of GIT-STORM and the individual contributions they make to the observed performance improvements.

Table 5.6: We report scores under visual inputs at 1M frames as well as aggregated human normalized mean and median. Following the conventions of [81], scores that are the highest or within 5% of the highest score are highlighted in bold.

Task	SAC	CURL	PPO	DrQ-v2	DreamerV3	STORM	GIT-STORM
Acrobot Swingup	5.1	5.1	2.3	128.4	**210.0**	12.2	2.1
Cartpole Balance	**963.1**	**979.0**	507.3	**991.5**	**996.4**	208.9	567.0
Cartpole Balance Sparse	**950.8**	**981.0**	890.4	**996.2**	**1000.0**	15.2	790.9
Cartpole Swingup	692.1	762.7	259.9	**858.9**	**819.1**	124.8	452.2
Cartpole Swingup Sparse	154.6	236.2	0.0	706.9	**792.9**	0.6	97.3
Cheetah Run	27.2	474.3	95.5	691.0	**728.7**	137.7	552.5
Cup Catch	163.9	**965.5**	821.4	**931.8**	**957.1**	735.5	841.5
Finger Spin	312.2	**877.1**	121.4	**846.7**	818.5	753.8	787.0
Finger Turn Easy	176.7	338.0	311.0	448.4	**787.7**	307.3	334.1
Finger Turn Hard	70.5	215.6	0.0	220.0	**810.8**	1.4	148.6
Hopper Hop	3.1	152.5	0.3	189.9	**369.6**	0.0	193.6
Hopper Stand	5.2	786.8	6.6	**893.0**	**900.6**	0.0	664.6
Pendulum Swingup	560.1	376.4	5.0	**839.7**	**806.3**	0.0	0.0
Quadruped Run	50.5	141.5	299.7	**407.0**	352.3	46.2	**396.6**
Quadruped Walk	49.7	123.7	107.1	**660.3**	352.6	55.4	445.4
Reacher Easy	86.5	609.3	705.8	**910.2**	**898.9**	72.7	222.4
Reacher Hard	9.1	400.2	12.6	**572.9**	499.2	24.3	12.3
Walker Run	26.9	376.2	32.7	517.1	**757.8**	387.2	427.6
Walker Stand	159.3	463.5	163.8	**974.1**	**976.7**	**934.8**	**954.8**
Walker Walk	38.9	828.8	96.0	762.9	**955.8**	758.0	854.7
Median	78.5	431.8	101.5	734.9	**808.5**	31.5	475.12
Mean	225.3	504.7	211.9	677.4	**739.6**	214.5	442.1

Figure 5.12 illustrates an ablation study on three Atari games (Hero, Freeway, and Boxing) and three DMC environments (Walker Walk, Walker Run, and Quadruped Run). Across both sets of environments, the removal of the MaskGIT head consistently results in poorer downstream performance (e.g., lower scores). Additionally, leveraging the dot product between ξ_t and MaskGIT embeddings has a substantial impact in environments such as Freeway, Walker Walk, and Quadruped Run. However, its influence appears negligible in other environments like Hero and Walker Run, suggesting that its efficacy may be context-dependent.

DIMENSIONS OF DYNAMIC PRIOR HEAD

In order to find the best configuration for the MaskGIT prior, we conduct experiments on three different environments with different embedding and vocabulary dimensions corresponding to the bidirectional transformer. While the performance of different configurations varies between environments, we find that a bigger embedding size achieves higher scores on average as seen in Figure 5.13.

As shown in DreamerV3 [26], the model achieves better performance as it increases in the number of trainable parameters. Thus, to provide a fair comparison with STORM, we restrict the transformer corresponding to the MaskGIT prior to a similar

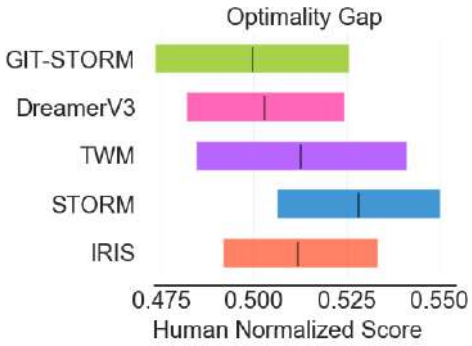


Figure 5.8: The Optimality Gap shows the amount by each algorithm fails to achieve human performance, where lower values are better.

number of parameters as the MLP prior defined in STORM.

VQ-VAE VS ONE HOT CATEGORICAL

The world model state in model-based RL is represented in terms of a latent representation based on raw observations from the environment. However, there is no clear consensus on the representation of the latent space, with SimPLE [253] using a Binary-VAE, IRIS [243] using a VQ-VAE while DreamerV3 [26], STORM [39] and TWM [246] employ a Categorical-VAE.

While recent methods show empirically the advantages of a Categorical-VAE in Atari environments, there is no comprehensive study on different latent space representations. Thus, Table 5.7 provides a comparison between a VQ-VAE and Categorical-VAE latent representation in the context of GIT-STORM, motivating our choice of latent space. The comparison is performed on three environments with different levels of complexity in terms of visual representations.

In order to keep the comparison between the two representations accurate, we scale down the VQ-VAE to only 32 codebook entries, each consisting of 32 dimensions, matching the size of the one-hot categorical representation of 32 categories with 32 classes each. While the VQ-VAE in IRIS [243] uses a considerably bigger

Table 5.7: Comparison between a VQ-VAE and Categorical-VAE latent representation for the world model state on three Atari 100k environments.

Game	VQ-VAE	One Hot Categorical
Boxing	0	**81**
Hero	0	**13351**
MsPacman	255	**1877**

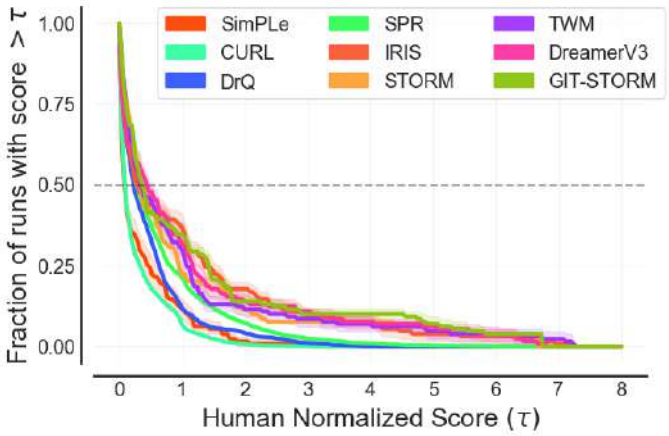


Figure 5.9: The graph presents the fraction of games above a certain human normalized score.

vocabulary and embedding size, we believe the additional number of parameters introduced provide a biased estimation of the representation capabilities of the latent space. Moreover, we notice that the VQ-VAE approach introduces a significant overhead in terms of training and sampling time. Table 5.7 shows that the VQ-VAE latent representations collapse and fail to learn a meaningful policy. In contrast the categorical representation achieves impressive results with the same compute budget.

STATE MIXER ANALYSIS

STATE MIXER INDUCTIVE BIASES

As described in Sec. 5.3.1, latent representations z_t and actions a_t are mixed using a state mixer function $g(\cdot)$. To understand the affect of different mixing strategies for the underlying task, we compare three different mixing functions in the DMC benchmark: (1) concatenation, (2) concatenation followed by attention and (3) cross attention between state and actions. Figure 5.15 illustrates the results. Surprisingly, we find that the simple approach works the best for the tasks – concatenation of state and action significantly outperforms the attention-based approaches in the chosen tasks.

STATE MIXER ABLATIONS

To evaluate the contribution of the State Mixer and its relevance compared to existing approaches, such as iVideoGPT [280], we conducted an ablation study. This analysis compares the effect of the State Mixer on downstream performance against the approach proposed in iVideoGPT. Figure 5.14 demonstrates that the State Mixer consistently outperforms the considered baselines. Interestingly, under the given setup, the iVideoGPT approach fails to learn meaningful policies. We hypothesize that this limitation arises from the scale of the training procedure and considered

environments. Specifically, iVideoGPT is designed to leverage much larger datasets, enabling it to learn robust representations.

Moreover, we observe that bypassing the State Mixer by directly concatenating and feeding state and action embeddings into the transformer allows the model to learn policies that are meaningful but perform suboptimally compared to the State Mixer-based approach. This finding highlights the effectiveness of the State Mixer in extracting and processing state-action representations crucial for learning optimal policies.

DYNAMICS HEAD ANALYSIS

KL DIVERGENCE COMPARISON

In this section, we present and analyse a comparison between our method and STORM in terms of the KL divergence of the dynamics module. Figure 5.16 illustrates the KL divergence loss for GIT-STORM and STORM across three environments: Hero, Boxing, and Freeway. It is evident that the KL divergence for GIT-STORM is consistently lower across all three environments, with a particularly significant difference observed in Boxing. This suggests that the dynamics module in GIT-STORM is better equipped to learn state transition dynamics compared to STORM, resulting in more accurate modeling of the underlying system dynamics.

DYNAMICS HEAD OUTPUT DISTRIBUTION VISUALIZATION

In this section, we inspect the output distributions of the dynamics head generated by the proposed GIT-STORM compared to those produced by STORM. Specifically, Figure 5.17 illustrates the mean probability distribution for generating a certain token at a given time step and frame. A closer examination of the density functions reveals that the mean distributions typically exhibit two peaks: one near zero, indicating that a given token does not need to be sampled, and a second, smaller peak, representing the confidence level for sampling a specific token.

The higher the second peak and the broader the distribution's support, the more confident the world model is in sampling tokens for a given dynamics state transition. Consistent with the perplexity values presented in Table 5.4, GIT-STORM produces more refined probability distributions, enabling it to make predictions with greater confidence compared to STORM.

VIDEO PREDICTION DOWNSTREAM TASK: TECO

In order to assess the capabilities of the MaskGIT prior in modelling latent dynamics across different tasks, we consider video generation tasks as a representative study. More specifically, we consider Temporally Consistent Transformer for Video Generation (TECO) [209] on DeepMind Lab (DMLab) [256] and Something-Something v.2 (SSv2) [257] datasets. TECO uses a spatial MaskGIT Prior to generate the state corresponding to the next timestep. Table 5.1 highlights the importance of the prior network and supports our earlier results on the Atari 100k benchmark. Indeed,

when replacing the MaskGit prior network with an MLP one with the same number of parameters, the FVD [188] on both DMLab and SSv2 datasets significantly increases, going from 48 to 153 and from 199 to 228 in the DMLab and SSv2 datasets respectively.

HYPERPARAMETERS

Table 5.8: Hyperparameters regarding the dynamics module, training settings and environment. We use the same hyperparameters as STORM [39] to focus our experiments on the MaskGIT prior.

Hyperparameter	Symbol	Value
Transformer layers	K	2
Transformer feature dimension	D	512
Transformer heads	-	8
Dropout probability	p	0.1
World model training batch size	B_1	16
World model training batch length	T	64
Imagination batch size	B_2	1024
Imagination context length	C	8
Imagination horizon	L	16
Update world model every env step	-	1
Update agent every env step	-	1
Environment context length	-	16
Gamma	γ	0.985
Lambda	λ	0.95
Entropy coefficiency	η	3×10^{-4}
Critic EMA decay	σ	0.98
optimiser	-	Adam [74]
World model learning rate	-	1.0×10^{-4}
World model gradient clipping	-	1000
Actor-critic learning rate	-	3.0×10^{-5}
Actor-critic gradient clipping	-	100
Gray scale input	-	False
Frame stacking	-	False
Frame skipping	-	4 (max over last 2 frames)
Use of life information	-	True
MaskGIT Transformer layers	-	4
MaskGIT Transformer feature dimension	-	128
MaskGIT Transformer heads	-	8
MaskGIT Dropout probability	-	0.0
Mask Schedule	-	cosine
Draft Rounds	T_{draft}	1
Revise Rounds	T_{revise}	1
Repetitions	M	1

Table 5.9: Specific structure of the image encoder used in GIT-STORM (ours) and STORM [39]. The size of the modules is omitted and can be derived from the shape of the tensors. ReLU refers to the rectified linear units used for activation, while Linear represents a fully-connected layer. Flatten and Reshape operations are employed to alter the indexing method of the tensor while preserving the data and their original order. Conv denotes a CNN layer [87], characterized by kernel = 4, stride = 2, and padding = 1. BN denotes the batch normalization layer [281].

Submodule	Output tensor shape
Input image (o_t)	$3 \times 64 \times 64$
Conv1 + BN1 + ReLU	$32 \times 32 \times 32$
Conv2 + BN2 + ReLU	$64 \times 16 \times 16$
Conv3 + BN3 + ReLU	$128 \times 8 \times 8$
Conv4 + BN4 + ReLU	$256 \times 4 \times 4$
Flatten	4096
Linear	1024
Reshape (produce $\mathcal{Z}_t$)	32×32

Table 5.10: Structure of the image decoder. DeConv denotes a transpose CNN layer [282], characterized by kernel = 4, stride = 2, and padding = 1.

Submodule	Output tensor shape
Random sample (z_t)	32×32
Flatten	1024
Linear + BN0 + ReLU	4096
Reshape	$256 \times 4 \times 4$
DeConv1 + BN1 + ReLU	$128 \times 8 \times 8$
DeConv2 + BN2 + ReLU	$64 \times 16 \times 16$
DeConv3 + BN3 + ReLU	$32 \times 32 \times 32$
DeConv4 (produce $\hat{o}_t$)	$3 \times 64 \times 64$

Table 5.11: Action mixer $\zeta_t = g_\theta(z_t, a_t)$. Concatenate denotes combining the last dimension of two tensors and merging them into one new tensor. The variable A represents the action dimension, which ranges from 3 to 18 across different games. D denotes the feature dimension of the Transformer. LN is an abbreviation for layer normalization [185].

Submodule	Output tensor shape
Random sample (z_t), Action (a_t)	32×32, A
Reshape and concatenate	$1024 + A$
Linear1 + LN1 + ReLU	D
Linear2 + LN2 (output e_t)	D

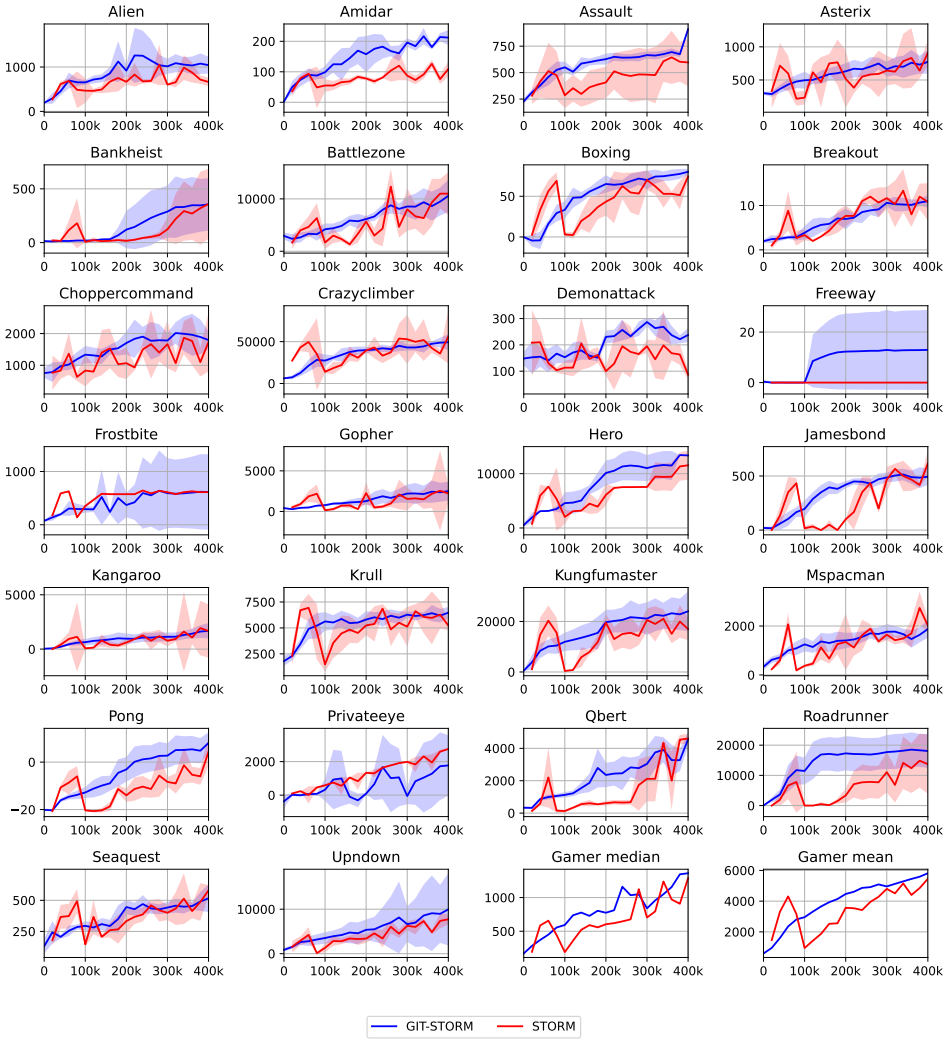


Figure 5.10: The solid line represents the average over 5 seeds while the fill area is defined in terms of maximum and minimum values corresponding to each checkpoint.

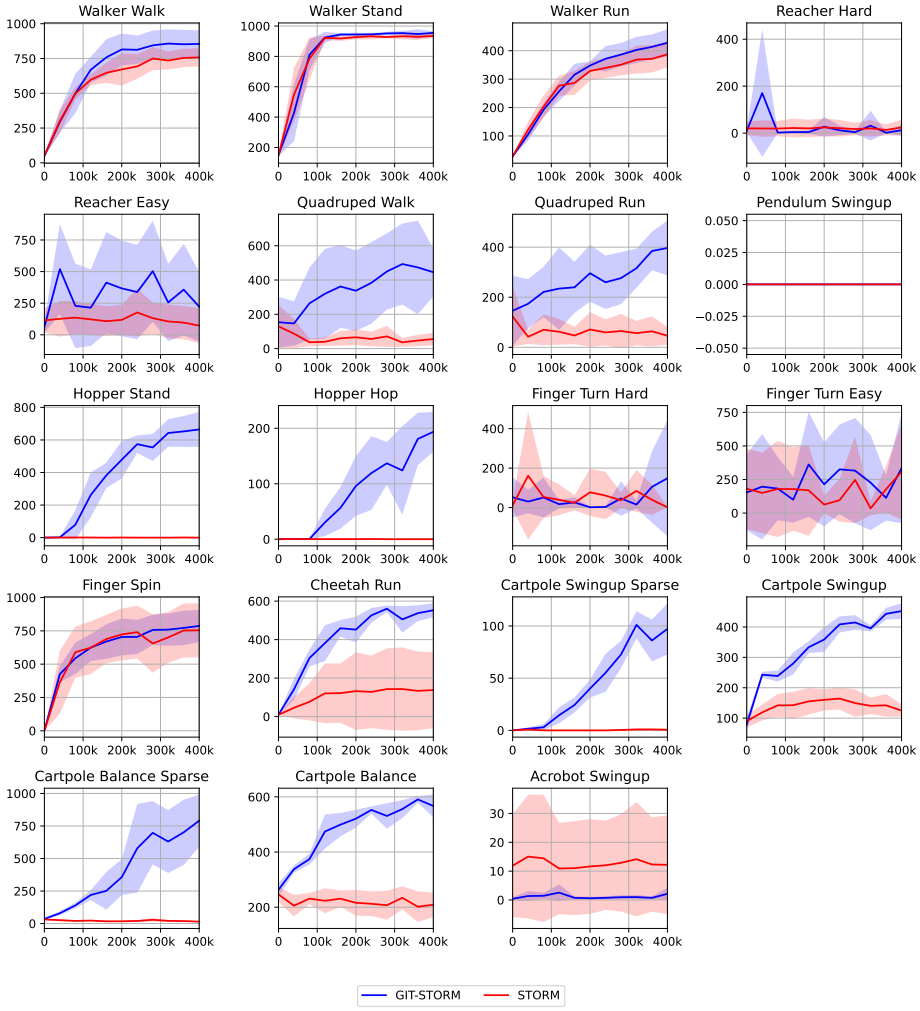


Figure 5.11: The solid line represents the average over 5 seeds while the fill area is defined in terms of standard deviation values corresponding to each checkpoint.

Table 5.12: $w_{1:T}$ is a learnable parameter matrix with shape $T \times D$, and T refers to the sequence length.

Submodule	Output tensor shape
Input ($e_{1:T}$)	
Add ($e_{1:T} + w_{1:T}$)	$T \times D$
LN	

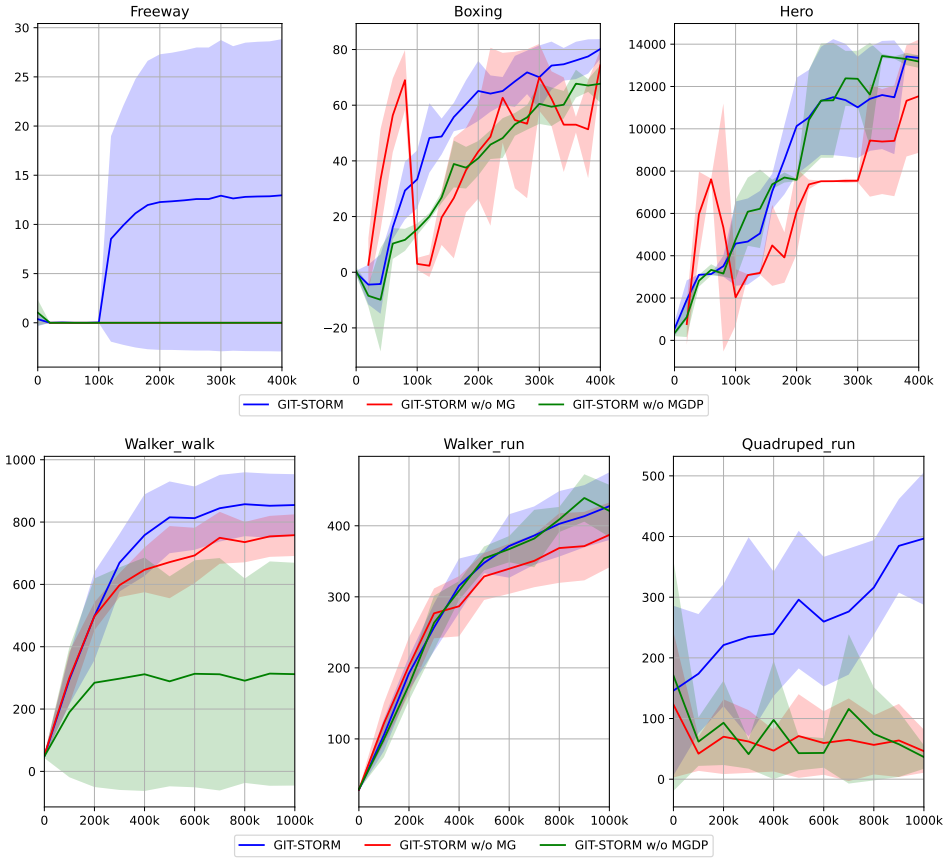


Figure 5.12: GIT-STORM ablation study on selected Atari and DMC environments: GIT-STORM w/o MG stands for without MaskGIT head, while GIT-STORM w/o MGDP stands for without MaskGIT dot product. All results are averaged across three random seeds.

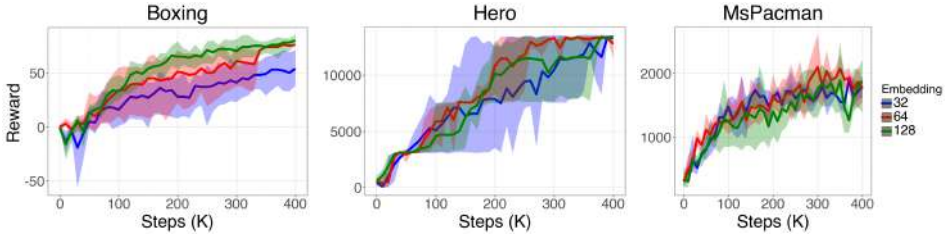


Figure 5.13: Different MaskGIT configurations for the Bidirectional Transformer embedding size. Bigger embedding sizes achieve better results. Three different seeds were used for this experiment.

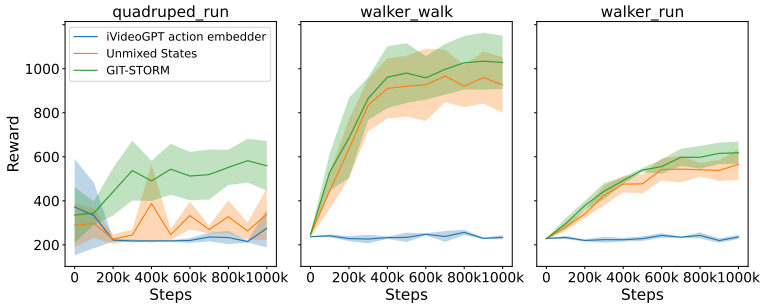


Figure 5.14: GIT-STORM action embedding approach ablation study on DMC environments. We consider: GIT-STORM, GIT-STORM using iVideoGPT action embedder and GIT-STORM without the State Mixer (labeled as Unmixed States). All results are averaged across three random seeds. GIT-STORM approach consistently outperforms the considered baselines.

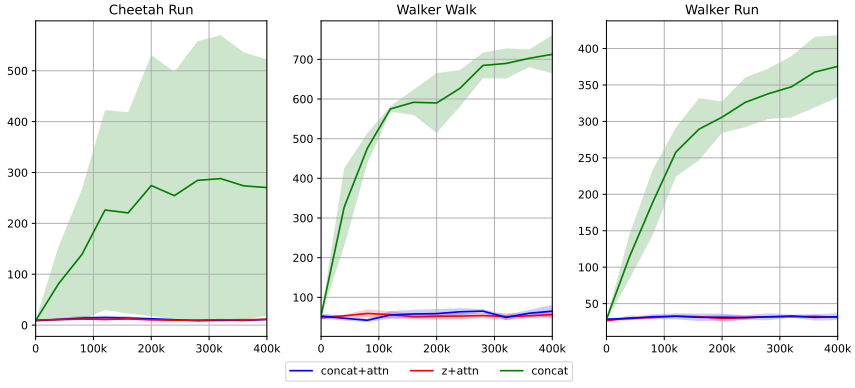


Figure 5.15: Comparison between different state and action mixing strategies tested in the DMC environments. All results are averaged across three random seeds. We find that simple concatination works the best for the chosen tasks.

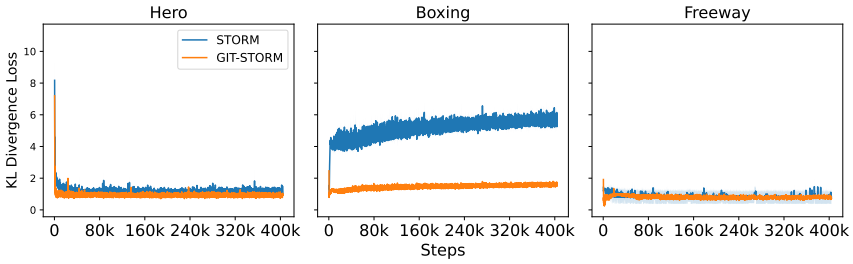


Figure 5.16: Comparison of GIT-STORM and STORM's KL divergence loss in Hero, Boxing and Freeway. GIT-STORM consistently presents a lower KL divergence. All results are averaged across three random seeds.

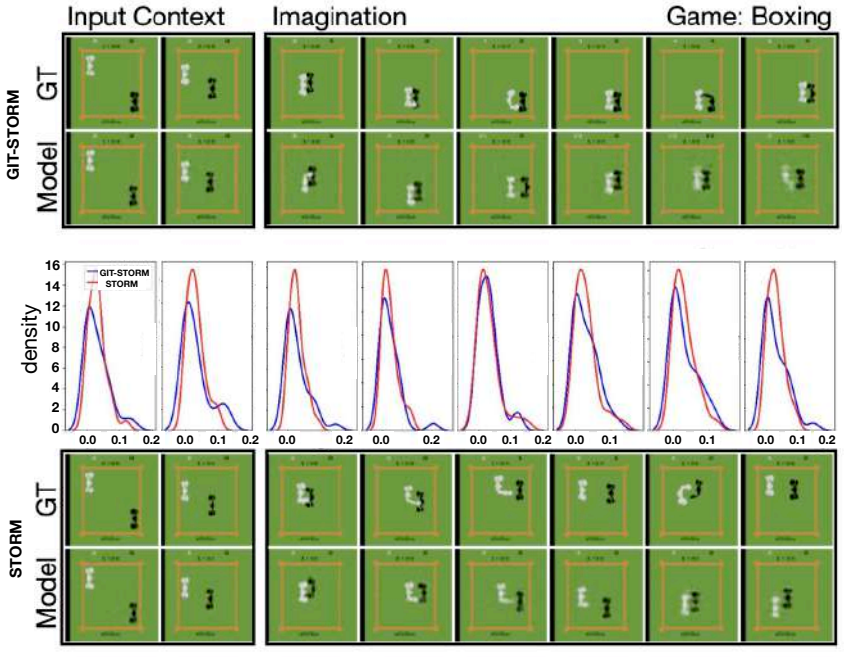


Figure 5.17: Above: GIT-STORM imagined trajectory in Boxing. Middle: Mean probability distribution of generating a certain token for a given time step and frame. Bottom: STORM imagined trajectory in Boxing.

Table 5.13: Dropout mechanism [283] can prevent overfitting.

Submodule	Module alias	Output tensor shape
Input features (label as x_1)		$T \times D$
Multi-head self attention Linear1 + Dropout(p) Residual (add x_1) LN1 (label as x_2)	MHSA	$T \times D$
Linear2 + ReLU Linear3 + Dropout(p) Residual (add x_2) LN2	FFN	$T \times 2D$ $T \times D$ $T \times D$ $T \times D$

Table 5.14: A 1-layer MLP corresponds to a fully-connected layer. 255 is the size of the bucket of symlog two-hot loss [26].

Module name	Symbol	MLP layers	Input/ MLP hidden/ Output dimension
Reward head	p_ϕ	3	D/ D/ 255
Termination head	p_ϕ	3	D/ D/ 1
Policy network	$\pi_\theta(a_t \mid s_t)$	3	D/ D/ A
Critic network	$V_\psi(s_t)$	3	D/ D/ 255

COMPUTATIONAL RESOURCES

Throughout our experiments, we make use of NVIDIA A100 and H100 GPUs for both training and evaluation on an internal cluster, a summary of which can be found in Table 5.15. For the Atari 100k benchmark, we find that each individual experiment requires around 20 hours to train. For the video prediction tasks, DMLab requires 3 days of training on 4 NVIDIA A100 GPUs. For DMC Vision tasks, we used H100 GPUs to sample from 16 environments concurrently, which reduced our training time to only 8 hours for 1M steps. Compared to this, using A100 for one environment takes 7 days. We acknowledge that the research project required more computing resources than the reported ones, due to preliminary experiments and model development.

Table 5.15

Experiment type	GPU Type	# of Days to train
Atari100k	1x A100	20 hours
DMLab	4x A100	3 days
DMC Vision	1x A100	8 hours

BASELINES

To assess our approach downstream capabilities on Atari 100k we select the following baselines: SimPLe [253] trains a policy using PPO [266] leveraging a world model represented as an action-conditioned video generation model; TWM [246] uses a transformer-based world model that leverages a Transformer-XL architecture and a replay buffer which uses a balanced sampling scheme [284]. IRIS [243], that uses a VideoGPT [60] based world model; DreamerV3 [26], a general algorithm which achieves SOTA results on a multitude of RL benchmarks. Lastly, we consider STORM [39], an efficient algorithm based on DreamerV3 that uses the transformer architecture for the world model. Since [26] shows that the replay buffer size is a scaling factor, to present a fair comparison we reproduce DreamerV3, which uses a replay buffer of 1M samples by default and full precision variables for the Atari 100k benchmark, using a replay buffer of $100K$ samples and half precision variables, consistent with our approach. Moreover, since STORM does not follow the evaluation protocol proposed in [267], after setting reproducible seeds, we reproduce STORM on the Atari 100k benchmark using the code released by the authors, and report the results as a result of running the released code.

For DMC Suite, we consider several state-of-the-art algorithms. Soft Actor-Critic (SAC) [264] is a popular algorithm for continuous control tasks, known for its data efficiency due to the use of experience replay. However, SAC often requires careful tuning, particularly for the entropy coefficient, and its performance can degrade when handling high-dimensional input spaces [285]. Another baseline is Proximal Policy Optimization (PPO) [266], a widely-used RL algorithm recognised for its robustness and stability across a range of tasks. Additionally, we include DrQ-v2 [286] and

CURL [43], both of which are tailored for visual environments. These methods leverage data augmentation to improve the robustness of learned policies, making them highly effective in scenarios where pixel-based observations dominate. Finally, we consider DreamerV3, which is the current state-of-the-art in this environment.

In order to meaninfgully evaluate the considered baselines we follow the protocol suggested in [267], which proposes the following metrics for a statistically grounded comparison:

- **Human Normalized Score**: To account for the discrepancies between raw score ranges in Atari games, and at the same time comparing the algorithm's capabilities with the human benchmark, the human normalized score is used to assess the performance of an algorithm on a specific environment. The Human Normalized Score is defined as $\frac{agent_{score} - random_{score}}{human_{score} - random_{score}}$.
- **Human Mean**: The Human Mean is an aggregate metric used to assess the performance across the whole Atari benchmark. The mean is computed using the Human Normalized Score for each environment, as previously defined.
- **Human Median**: Similar to the Human Mean, the Human Median is an aggregate metric across the Atari benchmark that is insensitive to high-score environments, which instead harm the statistical significance of the Human mean. According to [267], both the Human Mean and Human Median are necessary to assess the performance of an algorithm in Atari.
- **Interquantile Mean (IQM)**: Interquantile Mean is a popular statistical tool that only considers 50% of the results, effectively ignoring the lowest and highest performing environments. IQM aims to address the shortcomings of the Human Mean by ignoring outliers, while being more statistically significant than the Human Median, which only considers a single value.
- **Performance profiles (score distributions)**: Considering the variety of score ranges across different Atari environments, some of which may be heavy-tailed or contain outliers, point or interval estimates provide an incomplete picture with respect to an algorithm's performance. Performance profiles aim to alleviate this issues by revealing performance variability across tasks more significantly than interval and point estimates, like the Human Mean and Human Median.
- **Optimality Gap**: The Optimality Gap represents another alternative to the Human Mean, and accounts for how much the algorithm fails to meet a minimum Human Normalized Score of $\gamma = 1$. The metric considers γ as the desirable target and does not account for values greater than it. In the context of the Atari benchmark, $\gamma = 1$ represents the human performance. Using the Optimality Gap, the algorithms are compared without taking in consideration super-human performance, which is considered irrelevant.
- **Probability of Improvement**: Instead of treating algorithm's comparison as a binary decision (better or worse), the Probability of Improvement, indicates

a probability corresponding to how likely it is for algorithm X to outperform algorithm Y on a specific task.

For sequence modelling and video prediction task, we use the following metrics:

- **Perplexity**: Perplexity is mathematically defined as the exponentiated average negative log-likelihood of a sequence. Given a sequence of categorical representations $z_0, z_1, \ldots, z_t$, the perplexity of z is computed as:

$$\mathrm{PPL}(z) = \exp\left\{-\frac{1}{t}\sum_{i=1}^{t}\log p_\phi(z_i \mid z_{<i})\right\}.$$

  Here, $\log p_\theta(z_i \mid z_{<i})$ is the log-likelihood of the i-th token, conditioned on the preceding tokens $z_{<i}$, according to the model. In this context, perplexity serves as a measure of the model's ability to predict the tokenized representations of images in a sequence.

- **Fréchet Video Distance (FVD)**: Introduced in [188], FVD is a metric designed to evaluate the quality of video generation models. It builds on the idea of the widely-used Fréchet Inception Distance (FID), which is applied to assess the quality of generated images, but extends it to video by incorporating temporal dynamics. FVD is particularly effective for comparing the realism of generated videos with real video data, making it a crucial metric in video prediction and generation tasks.

6

STATEFUL ACTIVE FACILITATOR: COORDINATION AND ENVIRONMENTAL HETEROGENEITY IN COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Published at the Eleventh International Conference on Learning Representations (ICLR 2023)

Dianbo Liu
National University of Singapore, SG
dianbo.liu@nus.sg

Vedant Shah
Mila, University of Montreal, CA

Oussama Boussif
Mila, University of Montreal, CA

Cristian Meo
Delft University of Technology, NL

Anirudh Goyal
Mila, University of Montreal, CA

Tianmin Shu
MIT, US

Michael Mozer
Google Research, Brain Team, US

Nicolas Heess
DeepMind, US

Yoshua Bengio
CIFAR Chair; Mila, University of Montreal, CA

In cooperative multi-agent reinforcement learning, a team of agents works together to achieve a common goal. Different environments or tasks may require varying degrees of coordination among agents in order to achieve the goal in an optimal way. The nature of coordination will depend on the properties of the environment—its spatial layout, distribution of obstacles, dynamics, etc. We term this variation of properties within an environment as heterogeneity. Existing literature has not sufficiently addressed the fact that different environments may have different levels of heterogeneity. We formalize the notions of coordination level and heterogeneity level of an environment and present ***HECOGrid****, a suite of multi-agent RL environments that facilitates empirical evaluation of different MARL approaches across different levels of coordination and environmental heterogeneity by providing a quantitative control over coordination and heterogeneity levels of the environment. Further, we propose a Centralized Training Decentralized Execution learning approach called* ***Stateful Active Facilitator (SAF)*** *that enables agents to work efficiently in high-coordination and high-heterogeneity environments through a differentiable and shared knowledge source used during training and dynamic selection from a shared pool of policies. We evaluate SAF and compare its performance against baselines IPPO and MAPPO on HECOGrid. Our results show that SAF consistently outperforms the baselines across different tasks and different heterogeneity and coordination levels. We release the code for HECOGrid*[1] *as well as all our experiments.*

6.1. INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) studies the problem of sequential decision-making in an environment with multiple actors. A straightforward approach to MARL is to extend single agent RL algorithms such that each agent learns an independent policy [287]. [288] recently showed that PPO, when used for independent learning in multi-agent settings (called Independent PPO or IPPO) is in fact capable of beating several state-of-the-art approaches in MARL on competitive benchmarks such as StarCraft [289]. However, unlike most single-agent RL settings, learning in a multi-agent RL setting is faced with the unique problem of changing environment dynamics as other agents update their policy parameters, which makes it difficult to learn optimal behaviourpolicies. To address this problem of environment non-stationarity, a class of approaches called Centralized Training Decentralized Execution (CTDE) such as MADDPG [290], MAPPO [291], HAPPO and HTRPO [292] was developed. This usually consists of a centralized critic during training which has access to the observations of every agent and guides the policies of each agent. In many settings, MARL manifests itself in the form of cooperative tasks in which all the agents work together in order to achieve a common goal. This requires efficient coordination among the individual actors in order to learn optimal team behavior. Efficient coordination among the agents further aggravates the problem of learning in multi-agent settings.

Another challenge in practical multi-agent learning problems is *heterogeneity* in the environment. Environment heterogeneity in reinforcement learning has previously

[1] https://github.com/veds12/hecogrid and https://github.com/jaggbow/saf

Benchmark	Cooperative	Partial Obs.	Image Obs.	Coordination Control	Heterogeneity Control
SMAC	✓	✓	×	×	×
MeltingPot	✓	✓	✓	✓	×
MPE	✓	✓	×	×	×
SISL	✓	×	×	×	×
DOTA 2	✓	✓	×	×	×
HECOGrid	✓	✓	✓	✓	✓

Table 6.1: Comparison between our newly developed HECOGrid environments and widely used multi-agent reinforcement learning environments including SMAC [293], MeltingPot [294], MPE [290], SISL [295] and DOTA2 [72]

been studied in the context of federated learning in [296] which considers the problem of jointly optimising the behaviourof n agents located in n identical environments (same state space, same action space, and same reward function) with differing state-transition functions. However, in some real-world multi-agent learning problems, environment properties such as structure, dynamics, etc, may also vary within an environment, as compared to varying across different environments. Unmanned guided vehicles (UGVs) used for search and exploration may encounter different conditions, such as different distribution of obstacles or different terrains leading to differing dynamics in different regions. Warehouse robots coordinating to pick up a bunch of items might have to work in conditions varying from one section of the warehouse to another such as different organization of aisles. Similarly, as argued in [296], an autonomous drone should be able to adapt to different weather conditions that it encounters during its flight. We build upon the formulation of [296] to address the broader problem of heterogeneity within the environment.

We formally define two properties of an environment: **heterogeneity**, which is a quantitative measure of the variation in environment dynamics within the environment, and **coordination** which is a quantitative measure of the amount of coordination required amongst agents to solve the task at hand (we formally define **heterogeneity** and **coordination** in Section 6.3). The difficulty of an environment can vary based on the amount of heterogeneity and the level of coordination required to solve it. In order to investigate the effects of coordination and environmental heterogeneity in MARL, we need to systematically analyse the performance of different approaches on varying levels of these two factors. Recently, several benchmarks have been proposed to investigate the coordination abilities of MARL approaches, however, there exists no suite which allows systematically varying the heterogeneity of the environment. A quantitative control over the required coordination and heterogeneity levels of the environment can also facilitate testing the generalization and transfer properties of MARL algorithms across different levels of coordination and heterogeneity. A detailed analysis of the existing benchmarks can be found in Appendix 6.10

Previous MARL benchmarks have largely focused on evaluating coordination. As a result, while, there has been a lot of work which attempts addressing the

problem of coordination effectively, environment heterogeneity has been largely ignored. Heterogeneity so far has been an unintentional implicit component in the existing benchmarks. Hence, the problem of heterogeneity hasn't been sufficiently addressed. This is also apparent from our results where the existing baselines do not perform very competitively when evaluated on heterogeneity, since they were mainly designed to address the problem of coordination. Moreover, the fact that heterogeneity has been an unintentional implicit component of existing benchmarks, further strengthens our claim that heterogeneity is an essential and exigent factor in MARL tasks. Coordination and heterogeneity are ubiquitous factors for MARL. We believe that explicitly and separately considering these two as a separate factor and isolating them from other factors contributing to environment difficulty, will help motivate more research in how these can be tackled.

To address these limitations, we propose *HECOGrid*, a procedurally generated Multi-Agent Benchmark built upon MARLGrid [297]. HECOGrid consists of three different environments which allow the testing of algorithms across different coordination and environmental heterogeneity levels. Each environment consists of N agents and M treasures, where the goal is to maximise the total number of treasures picked up by the agents in one episode. c agents are required to pick up a single treasure, where c is the coordination level of the environment. The environment is spatially divided into h zones, and the environment dynamics vary from zone to zone. h is the level of heterogeneity of the environment. c, h, N, and M are controllable parameters. Table 6.1 presents a qualitative comparison between HECOGrid and other commonly used Multi-Agent RL environment suites.

HECOGrid also allows complete control over size of the map, number of obstacles, number of treasures, number of agents in addition to the coordination and heterogeneity levels. This provides ease of use of environments from small to very large scale with respect to the aforementioned parameters. This allows HECOGrid to be used as a standard challenging benchmark for evaluating not only coordination and heterogeneity but a lot of other factors. A lot of existing benchmarks [72, 289, 293] focus on moving away from toy-like grid world scenarios, using more complex scenarios with high dimensional observation and action spaces, continuous control, challenging dynamics and partial observability. Although HECOGrid has partial and image observations, it has a relatively small and discrete action space. Hence, HECOGrid cannot be used to test how an algorithm fares in continuous control and high dimensional action space scenarios. However, in most of existing standard benchmarks, it is non-trivial to modify environment parameters and hence it is difficult to perform a wide range of generalization and robustness studies. Melting Point [294] allows evaluating for out of distribution generalization, where the OOD scenarios can be defined by changing the *background population* of the environment. Unlike HECOGrid however, the physical layout of the environment (*substrate*) however, cannot be changed. HECOGrid, providing complete control over these environment parameters, make it easy to perform a wide range of experiments.

Further, to enable efficient training of MARL agents in high coordination and heterogeneity, we introduce a novel approach called **Stateful Active Facilitator** (SAF). SAF uses a shared *knowledge source* during training which learns to sift

through and interpret signals provided by all the agents before passing them to the centralized critic. In this sense, the knowledge source acts as an information bottleneck and helps implement a more efficient form of centralization by refining the information being passed to the critic. Further, recent work in modular deep learning [155, 298–300] has shown that different neural modules trained on a common objective lead to the emergence of specialist neural modules which help in improving performance via decomposition of the task. We hypothesize that a similar form of modularity can also be helpful in tackling the problem of heterogeneity. Instead of each agent using an individual monolithic policy, we propose the use of a *pool of policies* that are shared across different agents [298]. At each time step, each agent picks one policy from the pool which is used to determine its next action where the selection being conditioned on the current state of the agent. During execution the parameters of the shared pool of policies can be distributed to each agent which can then operate in a completely decentralized manner. Hence our method falls under the umbrella of Centralized Training Decentralized Execution (CTDE) methods.

Contributions. We introduce a set of cooperative MARL environments with adjustable coordination and heterogeneity levels. We also propose SAF- which consists of a shared knowledge source which is empirically shown to improve performance in high-level coordination settings, and a *pool of policies* that agents can dynamically choose from, which helps in tackling environmental heterogeneity. We show that the proposed approach consistently outperforms established baselines MAPPO [291] and IPPO [288] on environments across different coordination and heterogeneity levels. The knowledge source is the key to improved performance across different coordination levels whereas further ablation studies show that the pool of policies is the key to good performance across different levels of environmental heterogeneity.

6.2. RELATED WORK

Centralized Training Decentralized Execution (CTDE). These approaches are among the most commonly adopted variations for MARL in cooperative tasks and address the problem of environment non-stationarity in multi-agent RL. They usually involve a centralized critic which takes in global information, i.e. information from multiple agents, and decentralized policies whose learning are guided by the critic. [290] first proposed an extension of DDPG [301] to a multi-agent framework by using a shared critic and agent specific policies during training, and decentralized execution. [291] proposes the extension PPO [302] to a multi-agent framework in a similar manner. [292] extends trust region learning to a cooperative MARL setting in a way that the agents do not share parameters. [303] uses the standard centralized critic decentralized actors framework with a *counterfactual baseline*. [304] uses an information theory-based objective to promote novel behaviors in CTDE-based approaches. Value Decomposition [305–308], [306] approaches learn a factorized state-action value function. [305] proposes Value Decomposition Networks (VDN) which simply add the state-action value function of each agent to get the final state-action value function. [306] uses a mixing network to combine the action-value functions of each agent in a non-linear fashion.

Coordination in Multi Agent Reinforcement Learning. There have been several definitions of coordination in MARL, most of which come from the agents' perspective. [309] and [310] define coordination as the ability of agents to find optimal joint actions. [311] defines coordination as consensus among agents. [312] defines coordination as agents' ability to achieve a common goal. In contrast, we analyse coordination levels from the angle of the RL environment. Developing MARL algorithms that train coordinated policies requires sufficient exploration due to the presence of multiple equilbria. A large section of recent approaches revolves around explicitly taking into account the states and actions of other agents by learning differentiable communication channels between agents in order to train coordinated policies. [313] proposes DRQN-based communication protocols DIAL and RIAL. [314] proposes CommNet which uses a shared recurrent module which calculates the state of each agent conditioned on the previous state and the mean of messages received from all the other agents. [315] and [316] use attention based communication protocols where attention is used to transmit and integrate messages sent by other agents. Similar to our work, [317] attempts to learn coordinated behaviourin a CTDE setting without introducing explicit communication, by using the mutual information of the agents' actions.

Environmental Heterogeneity in MARL. Environmental heterogeneity is a relatively uncharted land in MARL and has only been explored to a very limited extent in RL as a whole. [296] analysed environmental heterogeneity in a federated learning setting. The authors define heterogeneity as different state transition functions among siloed clients in the federated system, while for each client the environment is homogeneous. In another more recent study by [318], heterogeneity of initial state distribution and heterogeneity of environment dynamics are both taken into consideration. Our problem is also closely related to Hidden Parameter Markov Decision Processes (HiP-MDPs) [319] which consider a set of closely related MDPs which can be fully specified with a bounded number of latent parameters. Our approach can also be seen as being related to the multi-task reinforcement learning setting, where the goal is to learn an optimal policy that can be generalised to a set of closely related tasks. In all of the above works, heterogeneity is considered to be arising from variations across different environments or tasks. In contrast, we focus on heterogeneity within an environment.

Extended related works can be found in appendix 6.10.

6.3. PRELIMINARIES

Notation. In this work, we consider a multi-agent version of *decentralized* Partially Observable Markov Decision Processes (Dec-POMDP) [320]. The environment is defined as $(\mathcal{N}, \mathcal{S}, \mathcal{O}, \mathbb{O}, \mathcal{A}, \mathcal{T}, \Pi, R, \gamma)$. $\mathcal{N} = \{1, ..., N\}$ denotes a set of $N > 1$ agents and $\mathcal{S}$ is the set of global states. $\mathcal{A} = A_1 \times \cdots \times A_N$ denotes the joint action space and $a_{i,t} \in A_i$ refers to the action of agent i at time step t. $\mathbb{O} = O_1 \times \cdots \times O_N$ denotes the set of partial observations where $o_{i,t} \in O_i$ stands for partial observation of agent i at time step t. The joint observation $o \in \mathbb{O}$ is given by the observation function $\mathcal{O} : (a_t, s_{t+1}) \rightarrow P(o_{t+1}|a_t, s_{t+1})$ where a_t, s_{t+1} and o_{t+1} are the joint actions, states

and observations respectively. Π is the set of policies available to the agents. To choose actions at timestep t, agent i uses a stochastic policy $\pi_{\theta_i}(a_{i,t}|h_{i,t})$ conditioned on its action-observation history $h_{i,t} = (o_{i,0}, a_{i,0}, \ldots, o_{i,t-1}, a_{i,t-1})$. Actions from all agents together produce the transition to the next state according to transition function $\mathcal{T} : (s_t, a_{1,t}, \ldots, a_{N,t}) \mapsto P(s_{t+1}|s_t, a_{1,t}, \ldots, a_{N,t})$. $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the global reward function conditioned on the joint state and actions. At timestep t, the agent team receives a reward $r_t = R(s_t, a_{1,t}, \ldots, a_{N,t})$ based on the current joint state s_t and the joint action $a_{1,t}, \ldots, a_{N,t}$. γ is the discount factor for future rewards.

Coordination level in cooperative MARL task. Here, we quantitatively define the coordination level required in a cooperative MARL task used in this study. Recall that $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the global reward function conditioned on the joint state and actions. At time step t, the agent team receives a reward $r_t = R(s_t, a_t)$ based on the current total state of all agents s_t and joint action a_t. Just as in the previous section, the joint state is factorized as follows: $s_t = (s_{e,t}, s_{1,t}, \ldots, s_{N,t})$ where $s_{e,t}$ is the state of the external environment and $s_{i,t}$ is the local state of agent i at timestep t and $i \in \mathcal{N}$.

Let $\mathcal{G} \subset \mathcal{N}$ denote a subset of $|\mathcal{G}| = k$ agents and different subsets can overlap. We define $R_{\mathcal{G}}(s_t, a_t) = R_{\mathcal{G}}(s_{e,t}, s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$ as the joint reward that can only be obtained when a subset of k agents cooperate, where $s_t^{\mathcal{G}} = \{s_{i,t}\}_{i \in \mathcal{G}}$ and $a_t^{\mathcal{G}} = \{a_{i,t}\}_{i \in \mathcal{G}}$. We can then write the joint reward as the sum of rewards contributed by all subset of agents:

$$R(s_t, a_t) = \sum_{\mathcal{G} \subset \mathcal{N}} R_{\mathcal{G}}(s_{e,t}, s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$$

Hence, the level of coordination c_t can be defined as the positive reward that can be obtained at time t if no less than c_t agents are involved in it:

$$c_t = \min_{k=1,\ldots,N} \{k | \exists \mathcal{G} \subset \mathcal{N} \text{ s.t } |\mathcal{G}| = k : \quad R_{\mathcal{G}}(s_{e,t}, s_t^{\mathcal{G}}, a_t^{\mathcal{G}}) > 0\}$$

Where $|\mathcal{G}|$ is the number of elements in $\mathcal{G}$. The global coordination level of the environment c can then simply be defined as: $c = \max_{t \geq 0}\{c_t\}$. This means that if there's at least one task in the environment that must be solved using the largest number of agents, then that number of agents (c_t) is defined as the coordination level of that environment.

It is worth mentioning the difference between the problem we explore and the formulation in previous studies such as [321]. We define coordination in a way that some rewards require at least k agents to coordinate with each other to obtain but different subsets of agents $\mathcal{G}_j$ do not have to be disjoint, i.e., one agent can be involved in obtaining more than one reward in a single time step.

Heterogeneity level of cooperative MARL task. Another aspect we like to explore is the heterogeneity of the RL environment. It is worth pointing out that the heterogeneity of the RL environment is different from the heterogeneity of agents or heterogeneity of policies as explored in previous studies ([311, 322].

For simplification, we define heterogeneity in a single-agent RL environment, which can be easily unwrapped into a multi-agent setting. We assume that the environment

has a collection of K different state-transition functions $\{\mathcal{T}_k : (s_t, a_t) \mapsto s_{t+1}\}_{1 \leq k \leq K}$. At each timestep t, whenever the agent takes an action, its next state is governed by one of the K state-transition functions, and that choice is decided according to some (possibly latent) variable ν_t. K is then defined as the level of heterogeneity of the environment, if $K = 1$ then the environment is said to be homogeneous. In this study, we implement ν_t as the position of the agent in the environment, which means the state-transition function depends on where the agent is in the environment.

6.4. HECOGRID: MARL ENVIRONMENTS FOR VARYING COORDINATION AND ENVIRONMENTAL HETEROGENEITY LEVELS

We introduce a set of three cooperative MARL environments, which we collectively call HECOGrid, that allows manipulation of coordination and environmental heterogeneity levels in addition to several other properties. HECOGrid consists of `TeamSupport`, `TeamTogether` and `KeyForTreasure` environments as shown in Fig. 6.1. In this section, we describe each HECOGrid environment in detail.

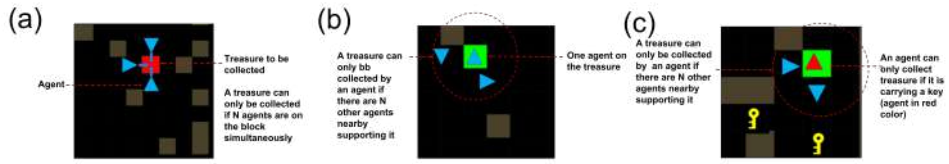


Figure 6.1: Three cooperative multi-agent reinforcement learning environments developed in this study that allow quantitative control of coordination and environmental heterogeneity levels to adjust difficulty of cooperative tasks: (a) `TeamTogether` environment, (b) `TeamSupport` environment and (c) `KeyForTreasure` environment.

TeamTogether Environment. The cooperative team task in this environment is to collect as many treasures as possible in a limited number of time steps. Each treasure is presented as a bright box in the environment and becomes grey once collected. In order for a treasure to be collected, a certain number of agents need to step onto it simultaneously. The number of agents required for collection is the level of coordination of the task.

TeamSupport Environment. This environment is similar to `TeamTogether` except that in order for an agent to collect a treasure, instead of being on the treasure together simultaneously with other agents, it needs to step onto the box and with a certain number of agents within a fixed distance (set to 2 by default) to support the collection. This number of agents required for collection support (including the agent that actually collects) is defined as the level of coordination of the task. Rewards are distributed equally across all agents in the whole team.

KeyForTreasure Environment. This environment is similar to `TeamSupport`

except that an agent can only collect a treasure if it is carrying a key, and to collect the treasure, a certain number of agents need to be on the box simultaneously. This additional key-searching step increases the difficulty of the task. If an agent picks up a key, its color changes.

In all the environments, all the rewards are distributed equally across all agents on the team.

Environmental heterogeneity in HECOGrid. We implement environmental heterogeneity by dividing the grid into K zones. For each zone, the transition function $\mathcal{T}$ is different. Concretely, each action leads to a different state depending on which zone the agent is in (e.g. action number 1 may make the agent turn left, right, move forward or perform some other action depending on what zone of the grid the agent is present in).

6.5. SAF: The Stateful Active Facilitator

In this section we describe the proposed method that consists of two main components: a Knowledge Source (KS) that enhances coordination among the agents and a policy pool (PP) that allows agents to dynamically select a policy, enabling agents to exhibit diverse behaviors and have distinct goals, as well as handling heterogeneous environments.

After receiving the local observations, the agents produce messages conditioned on the observations and send them to the KS. These messages are integrated into the KS via a soft attention mechanism. This enables the KS to sift through the information shared by the agents and filter out the irrelevant information, before sending it back to the agents. The agents then utilise this message to define a more informative state which is then used by the critic. Hence, the KS acts as an information bottleneck, which by filtering out irrelevant information, aids the centralized critic in coordinating the agents better. Further, each agents dynamically selects a policy from the shared pool of policies conditioned on its current state. The current state of an agent is simply an encoding of the local observation received by the agent. By using a pool of policies, we aim to train specialists which are suited to tackle different environmental conditions, hence aiding in tackling heterogeneity.

Technically, SAF can be obtained by augmenting any of the existing Centralized Training Decentralized Execution (CTDE) with the KS and PP. The setup used in our experiments closely resembles that of MAPPO [291] with the centralized critic augmented with the KS and the policies of agents replaced by a shared pool of policies. SAF is trained in the same manner as MAPPO, where the centralized critic and the KS are used for guiding the policies. These are not used during execution, which allows is to retain the CTDE nature. We train SAF end to end by using the same loss function and standard implementation practices as discussed in MAPPO in [291].

Step 1: Generating the messages. Each agent i receives a partial observation $o_{i,t}$ at each time step t. These observations are encoded into messages which are written into the KS by a common encoder g_θ: $m'_{i,t} = g_\theta(o_{i,t})$; $m'_{i,t} \in \mathbb{R}^{d_m}$. We denote the set of messages generated by the agents at time step t by $\mathbf{M}'_t$:

$$\mathbf{M}'_t = \{m'_{i,t} | 1 \leq i \leq N\}$$

Step 2: Writing into the Knowledge Source. The messages $\mathbf{M}'_t$ generated in step one are distilled into a latent state which we term as a **Knowledge Source** or KS. We represent the KS state at time step t by $\mathbf{F}_t$. $\mathbf{F}_t$ consists of L slots $\{l_0, l_1, ..l_{L-1}\}$, each of dimension d_l so that $\mathbf{F}_t \in \mathbb{R}^{L \times d_l}$.

The messages in $\mathbf{M}'_t$ compete with each other to write into each KS's state slot via a cross-attention mechanism. The query, in this case, is a linear projection of the $\mathbf{F}_t$, i.e., $\widetilde{\mathbf{Q}} = \mathbf{F}_t \widetilde{\mathbf{W}}^q$, whereas the keys and values are linear projections of the messages $\mathbf{M}'_t$. KS state is updated as:

$$\mathbf{F}_t \leftarrow \text{softmax}\left(\frac{\widetilde{\mathbf{Q}}(\mathbf{M}'_t \widetilde{\mathbf{W}}^e)^{\mathrm{T}}}{\sqrt{d_e}}\right) \mathbf{M}'_t \widetilde{\mathbf{W}}^v$$

After this, self-attention is applied to the KS using a transformer encoder tower constituting a `Perceiver-IO` architecture [323].

Step 3: Reading from the Knowledge Source. The KS makes the updated state available to the agents should they deem to use it. We again utilise cross attention to perform the reading operation. All the agents create queries $\mathbf{Q}^s_t = \{q^s_{i,t} | 1 \leq i \leq N\} \in \mathbb{R}^{N \times d_e}$ where $q^s_{i,t} = \mathbf{W}^q_{\text{read}} s_{i,t}$ and $s_{i,t} = g_\omega(o_{i,t})$ are encoded partial observations . Generated queries are matched with the keys $\boldsymbol{\kappa} = \mathbf{F}_t \mathbf{W}^e \in \mathbb{R}^{L \times d_e}$ from the updated state of SAF. As a result, the attention mechanism can be written as:

$$\mathbf{M}_t = \text{softmax}\left(\frac{\mathbf{Q}^s_t \boldsymbol{\kappa}^T}{\sqrt{d_e}}\right) \mathbf{F}_t \mathbf{W}^v \tag{6.1}$$

where $\mathbf{M}_t = \{m_{i,t} | 1 \leq i \leq N\}$. Consequently, the read messages are used to define a more informative state $s^{\text{SAF}}_{i,t} = g_\phi([s_{i,t}, m_{i,t}])$, where g_ϕ is parameterized as a neural network. Finally, the new state $s^{\text{SAF}}_{i,t}$ is used by the critic to compute values. Interestingly, since $s^{\text{SAF}}_{i,t}$ is exclusively used by the critic, which is only used during training, SAF do not uses communication during execution.

Step 4: Policy Selection. In order to perform policy selection for each policy we define an associated signature key which is initialized randomly at the start of the training: $\mathbf{k}_\Pi = \{k_{\pi^u} | 1 \leq u \leq U\}$. These keys are matched against queries computed as deterministic function of the encoded partial observation $q^{\text{policy}}_{i,t} = g_{\text{psel}}(s_{i,t})$, where g_{psel} is parametrized as a neural network.

$$\text{index}_i = \text{GumbelSoftmax}\left(\frac{q^{\text{policy}}_{i,t}(\mathbf{k}_\Pi)^T}{\sqrt{d_m}}\right) \tag{6.2}$$

As a result of this attention procedure, agent i selects a policy π^{index_i}. This operation is performed independently for each agent, i.e. each agent selects a policy from the policy pool. Therefore, it does not involve communication among different agents.

6.6. EXPERIMENTS

In this section, we design empirical experiments to understand the performance of SAF and its potential limitations by exploring the following questions: (a) how much difficulty do high levels of coordination and environmental heterogeneity cause to cooperative MARL tasks? (b) does SAF perform well when coordination or/and heterogeneity levels are high? (c) Is SAF robust to changes of coordination and heterogeneity levels? and (d) SAF has two components. How does each component contribute to the performance at high coordination and heterogeneity levels?

Baseline Methods. We compare SAF with two widely used algorithms with related architectural designs and similar number of parameters to SAF, namely Independent PPO (IPPO) [288] and multi-agent PPO (MAPPO) [291] (Table 6.2). In IPPO, each agent has its own actor and critic and does not share information with other agents. In MAPPO, instead of being trained in a decentralized manner, the critic takes information from all agents in each step as inputs during training, and agents operate in a decentralized manner without sharing information during execution. Since, the agents in our environments are homogeneous, we use the parameter sharing for MAPPO, where the the actor and critic parameters are shared across all agents [324, 325]. SAF has similar training and execution strategy as MAPPO but uses an added component - the KS before passing information to the critic during training, and a shared pool of policies instead of a single shared policy for each agent.

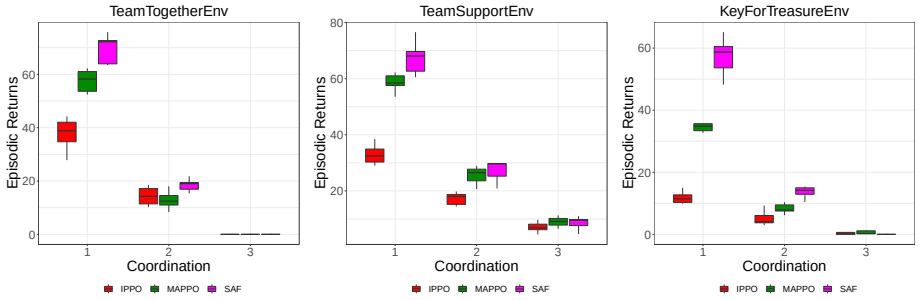


Figure 6.2: Test-time results for SAF, MAPPO and IPPO on `TeamTogether`, `TeamSupport` and `KeyForTreasure` environments on varying levels of coordination. The heterogeneity level is fixed at 1. Performance of all algorithms decreases as coordination levels increase with SAF showing better performance across all environments.

High levels of coordination and environmental heterogeneity. To understand how much difficulty high levels of coordination in the environment cause, we conducted experiments in all three HECOGrid environments for coordination levels 1 to 3 with heterogeneity level set to 1. We train all methods for 10M steps for all experiments. Our results, as shown in Figure 6.2 show that the performance of all three methods decreases dramatically as coordination levels increase. Performance of all three methods show a more than 50% decrease in performance at coordination

level 2, as compared with coordination level 1, in all environments. At a coordination level of 3, all methods fail to show meaningful behavior. These observations indicate that tasks requiring more agents to work together for reward collection are extremely challenging in a cooperative MARL setting.

To understand how much difficulty high levels of environmental heterogeneity cause, we conducted experiments in all three HECOGrid environments for heterogeneity levels 1 to 5 with coordination level set to 1. All methods show a decrease in performance as the environments become more heterogeneous, though to a smaller extent as compared with coordination levels (see Figure 6.3). We provide further results for experiments performed in cases where coordination and heterogeneity levels are high simultaneously in Appendix 6.10.

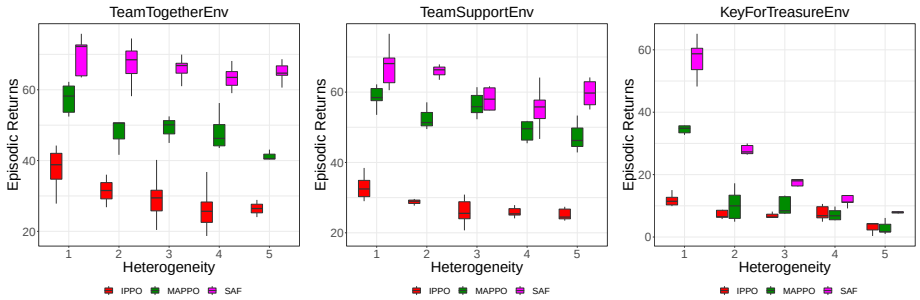


Figure 6.3: Test-time results for SAF, MAPPO and IPPO on `TeamTogether`, `TeamSupport` and `KeyForTreasure` environments on varying levels of heterogeneity. The coordination level is fixed at 1. All algorithms show decreased performance as heterogeneity increases. SAF shows better performance in more cases.

Contribution of each component of saf. In the method section, we design SAF by hypothesizing that in order to tackle coordination and environment heterogeneity, two key elements are necessary: the use of a shared knowledge source (KS) and a shared pool of policies (PP) from which the agents can dynamically choose. We wish to understand how much each component contributes to the performance of SAF in different scenarios. To investigate this question, we conduct experiments using different ablated versions of SAF in cooperative tasks with different levels of heterogeneity and coordination. As seen in Figure 6.4 in the Appendix, our experimental results indicate that the knowledge source contributes to the performance of SAF in all cooperative tasks while the shared pool of policies significantly improves the performance of the agents in heterogeneous environments and has minimal contribution to tasks requiring high coordination.

6.7. CONCLUSION

In this work, we explore coordination and heterogeneity levels of cooperative MARL environments by developing a set of environments, HECOGrid, which allows full

quantitative control over coordination and heterogeneity levels. Moreover, we propose a novel algorithm that enables agents to perform well in difficult environments with high levels of coordination and heterogeneity. Our experimental results suggest that high coordination and heterogeneity do make cooperative tasks challenging and our SAF method allow agents to gain better performance in these environments.

6.8. ETHIC STATEMENT AND REPRODUCIBILITY

To the best of the authors' knowledge, this study does not involve any ethical issues.The authors aim to maximise the reproducibility of the study. The codes of this project including the new environment constructed will be released in the camera-ready version. In the methods section, notions align with existing literature. A detailed description of each step in the SAF algorithm is given in the method section and a full algorithm is provided in the appendix.

6.9. ACKNOWLEDGEMENT AND AUTHOR CONTRIBUTION

Y.B. supervised the project and contributed to conceptualization, presentation and manuscript writing. N.H. contributed to the conceptualization and experimental design. M.M. contributed to conceptualization, experimental design and manuscript writing. T.S. contributed to reinforcement learning task design. A.G. contributed to project initialization, conceptualization, coordination, experimental design, method development and manuscript writing. C.M. contributed to implementation, experimental design, method development and manuscript writing. D.L. , V.S. and O.B. contributed to project initialization, conceptualization, implementation, experimental design, method development and manuscript writing.

In addition, we thank CIFAR, Mila - Quebec AI institute, University of Montreal, BITS Pilani, TUDelft, Google, Deepmind, MIT, and Compute Canada for providing all the resources to make the project possible.

6.10. APPENDIX

ADDITIONAL RELATED WORK

Information Bottleneck. With the emergence of modular deep learning architectures [5, 155, 326–329] which require communication among different model components, there has been a development of methods that introduce a bottleneck in this communication to a fixed bandwidth which helps to communicate only the relevant information. [330] use a VQ-VAE [57] to discretize the information being communicated. Inspired by the theories in cognitive neuroscience [56, 331, 332], [333] proposes the use of a generic *shared workspace* which acts as a bottleneck for communication among different components of multi-component architectures and promotes the emergence of *specialist* components. We use SAF, which is similar to the shared workspace that different agents compete to write information to and read information from.

Communication in MARL. Communication involves deciding which message to be shared and determining how the message-sending process is implemented. [313] and [314] implemented learnable inter-agent communication protocols. [315] first proposed using attention for communication where attention is used for integrating the received information as well as determining when communication is needed. [316] uses multiple rounds of direct pairwise inter-agent communication in addition to the centralized critic where the messages sent by each agent are formed by encoding its partial observation, and the messages received by each agent are integrated into its current state by using a soft-attention mechanism. [334] uses intentions represented as encoded imagined trajectories as messages where the encoding is done via a soft-attention mechanism with the messages received by the agent. [335] trains a model for each agent to infer the intentions of other agents in a supervised manner, where the communicated message denotes the intentions of each agent. The above-mentioned approaches require a computational complexity that is quadratic in the number of agents whereas our approach has a computational complexity that is linear in the number of agents. Moreover, we show that our approach is able to outperform several standard baselines using messages which can be computed as simply encoding each agent's partial observation. [336] developed a transformer-based multi-agent reinforcement learning method that models MARL decision-making as a sequential model.

Coordination in MARL and the Pareto-optimal Nash equilibrium. In the field of MARL, coordination is usually defined as the ability for agents to make optimal decisions to achieve a common goal by finding an optimal joint action in a dynamic environment[311, 312]. One way to find the optimal joint actions by a group of agents is by studying the Pareto-optimal Nash equilibrium [337], which describes the optimal solution as one in which no agent's expected gain can be further increased without compromising other agents' gain. However, there exist several challenges in cooperative MARL systems to achieve Pareto-optimal solutions. In the following sections, we are going to explain three of these challenges, which are the ones we seek to tackle in this study as well as their links to coordination and environmental heterogeneity levels.

Environmental heterogeneity and the non-stationarity problem In MARL,

the transition probabilities associated with the action of a single agent change over time as the action dynamics of the other agents change [338]. To solve this problem of non-stationarity, most recent MARL methods follow the Centralized Training Decentralized Execution paradigm. The most extreme case of centralized training is when all agents share the same set of parameters. However, parameter sharing also assumes that all agents have the same behaviors, which is not true when there is heterogeneity either among the agents themselves or in the environment. Previous studies use indicator-based methods to personalize a shared policy in a group of heterogeneous agents[325], however, environmental heterogeneity has been less explored in the literature [296].

Environmental heterogeneity and the alter-exploration problem Another problem environmental heterogeneity may cause in cooperative MARL is the alter-exploration problem. The balance between exploration and exploitation is crucial for all reinforcement learning tasks. In cooperative MARL this problem arises when exploration of one agent may penalize other agents and their corresponding policies during training as the cooperative agents share rewards[339, 340]. Environmental heterogeneity could potentially lead to worse alter-exploration problems as there tend to be more unseen states for an exploring agent which may result in higher and more frequent penalties. In this study, we seek to solve the above-mentioned problem using a combination of inter-agent communication via an active facilitator and a shared pool of policies.

Existing Environments in MARL. Some of the existing benchmarks based on online multiplayer games, attempt to move away from the toy-like grid world setting for MARL environments in favor of more realistic environments, by making use of high-dimensional observation and action spaces, continuous action spaces, challenging dynamics, and partial observability [72, 293, 294, 341]. These benchmarks focus on decentralized control in cooperative tasks and agents are heterogeneous (i.e., different types of agents having different abilities). In principle, it is possible to vary the levels of coordination and heterogeneity since the difficulties of different environments vary. However, there is no well-defined notion of the two concepts and these can't be varied in a controlled fashion. MeltingPot, which was recently proposed in [294] focuses on test-time generalization abilities of a group of agents includes a wide range of scenarios: competitive games, games of pure common interest, team-based competitive games, and mixed motion games which stress test the coordination abilities of the agents. However, similar to other benchmarks, there is no systematic decomposition nor a quantitative notion of the concepts of *coordination* and *environmental heterogeneity*.

ADDITIONAL RESULTS

In this section, we show the training curves for SAF, MAPPO and IPPO on `KeyForTreasure`, `TeamSupport` and `TeamTogether` environments. We additionally present more ablation results in the Out-of-Distribution setting.

Performance when increasing coordination or/and heterogeneity levels. SAF shows significant performance improvement upon MAPPO and IPPO at coordination levels 1 and 2 (Figure 6.2). In addition, SAF shows faster performance increase

at an early stage of the training process (figure 6.5). This suggests potential advantages for training agents using SAF in cooperative tasks requiring high coordination. At most heterogeneity levels, SAF shows performance improvement upon MAPPO and IPPO in all HECOGrid environments and a faster increase in performance at early stages of the training (Figure 6.3 and 6.5). This suggests potential advantages for training agents using SAF in a cooperative environment which are heterogeneous. In addition to manipulating coordination and heterogeneity levels separately, experiments are conducted to understand if SAF can perform well in environments in which both parameters are high. In the relatively easy `TeamSupport` environment with both coordination and heterogeneity set at 2, 3 and 4, SAF again shows improved performance over IPPO and MAPPO (Figure 6.5(c)). Figure 6.5(a) shows the training curves for SAF, MAPPO, and IPPO on `KeyForTreasure`, `TeamSupport` and `TeamTogether` environments for a coordination level of 2. It shows a gap in performance between SAF and the baselines and this gap is further enlarged when it comes to a heterogeneity level of 2 (see Figure 6.5(b)) which shows that SAF is effectively able to handle changes in the environment's dynamics.

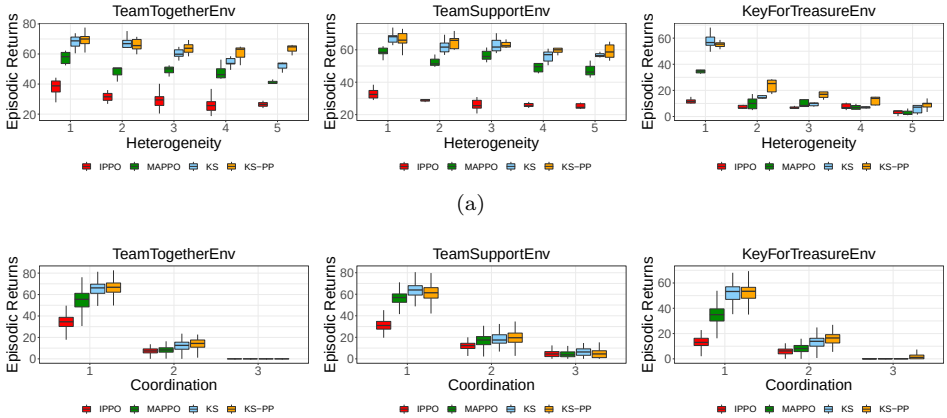


Figure 6.4: Ablation study to understand the contribution of the Knowledge Source (KS) and the shared policy pool (PP) to the performance of SAF in HECOGrid environments. In the legend, *KS* indicatess SAF without the Pool of Policies whereas *KS-PP* essentially means SAF (a) Performance of the KS and the pool of policies against baselines in increasing levels of heterogeneity. (b) Performance of the KS and the pool of policies against baselines in increasing levels of coordination. KS contributes to performance in all settings and PP especially improves performance in heterogeneous environments.

Robustness to changes in coordination and heterogeneity levels. In most real-world applications, such as robots in warehouses, the coordination levels as well as environmental heterogeneity levels can change over time and may even be unknown

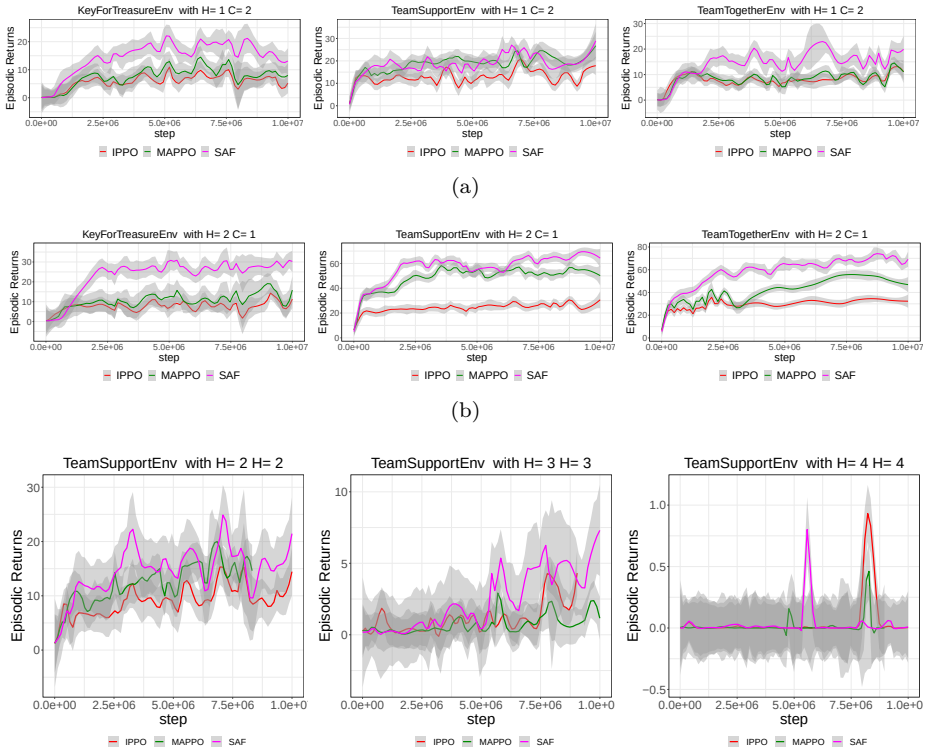


Figure 6.5: Examples of training curves for SAF, MAPPO and IPPO on the `KeyForTreasure`, `TeamSupport` and `TeamTogether` at different coordination and Heterogeneity levels. At the initial stage of the SAF show a faster increase in performance. After convergence, SAF shows improved performance in most tasks compared to IPPO and MAPPO.

to the agents. Therefore, the agents' robustness to such changes is important. To understand if agents trained with SAF can still function well in these out-of-distribution (OOD) settings, we conduct experiments to test the agents' performance on `TeamSupport` and `TeamTogether` environments with heterogeneity or coordination levels that are different as compared to the ones used during training. First, we train the agents in environments with a coordination level of 2 and a heterogeneity level of 1 and test their performance at coordination levels between 1 and 3, and a heterogeneity level of 1. As shown in Figure 6.6, SAF shows better transfer in the more difficult `TeamTogether` environment than other methods but fails to perform as well as MAPPO in the `TeamSupport` environment. Next, we train the agents in environments with a coordination level of 1 and a heterogeneity level of 2 and test their performance at a coordination level of 1 and heterogeneity levels between 1 and 5. As shown in figure 6.6, SAF shows superior performance in `TeamTogether` environment and

matches the performance of MAPPO in `TeamSupport` environments. This suggests that SAF has similar robustness to changes in coordination and heterogeneity levels as some of the widely used baselines in the MARL community.

Ablation Study for OOD generalization Figure 6.7(a) shows test-time generalization results on the `TeamTogether` and `TeamSupport` environments where the training coordination level was set to 2 and the heterogeneity was set to 1. The pool of policies in SAF is important in getting good performance especially when it's tested on levels of coordination not seen during training. Moreover, Figure 6.7(b) further validates that the pool of policies is important in handling varying environment dynamics as SAF was trained on a heterogeneity level of 2 and a coordination level of 1 and the results on unseen levels of heterogeneity are better than SAF trained without a pool of policies. These ablations show that the introduced pool of policies in SAF is key to its performance.

Comparison with QPLEX baseline Performance of QPLEX and QPLEX with a shared pool of policies are compared in different environments with 5 agents. The reason only pool of policies but not share knowledge source was used is because QPLEX already has a similar mechanism. The results suggest that a shared pool of policies among agents improve learning efficiency when coordination level is high (Figure 6.8).

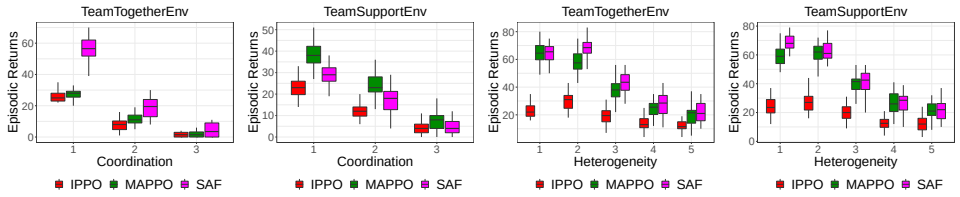
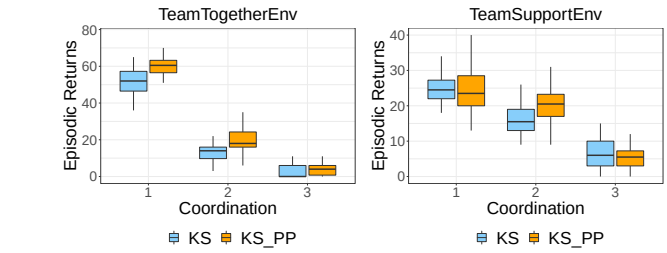


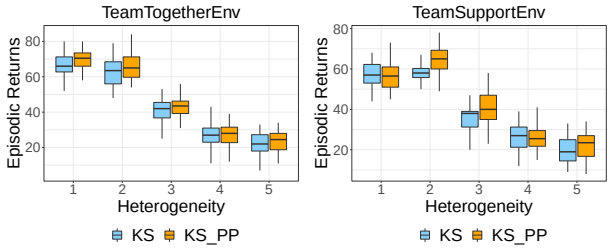
Figure 6.6: Out-of-Distribution generalization study to understand the robustness of SAF to changes in coordination and heterogeneity levels in HECOGrid environments. Agents are trained at certain heterogeneity and coordination levels, and tested on unseen levels. In general, SAF matches MAPPO in robustness to shifts in coordination or heterogeneity level

Method	No. of parameters
MAPPO	2,477,296
IPPO	1,740,016
saf (Our Method)	2,698,342

Table 6.2: Comparing the number of parameters used in our implementations of the discussed approaches



(a)



(b)

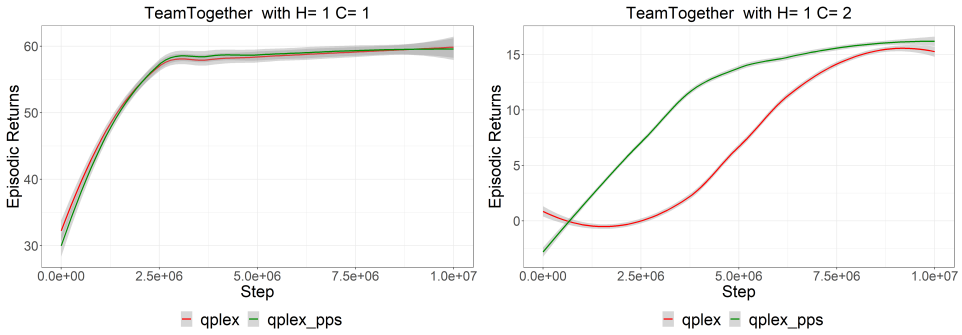
Figure 6.7: Ablation study for Out-of-Distribution generalization. In order to understand the robustness of different ablated models to shifts in either coordination or heterogeneity levels, models trained with coordination level 2 and heterogeneity level 1 are tested across different coordination levels (Figure 6.7(a)). In a similar manner, The models trained on coordination level 1 and heterogeneity level 2 are tested across different heterogeneity levels (Figure 6.7(b)). *KS* indicates SAF without pool of policies whereas, *KS_PP* means SAF.

IMPLEMENTATION DETAILS

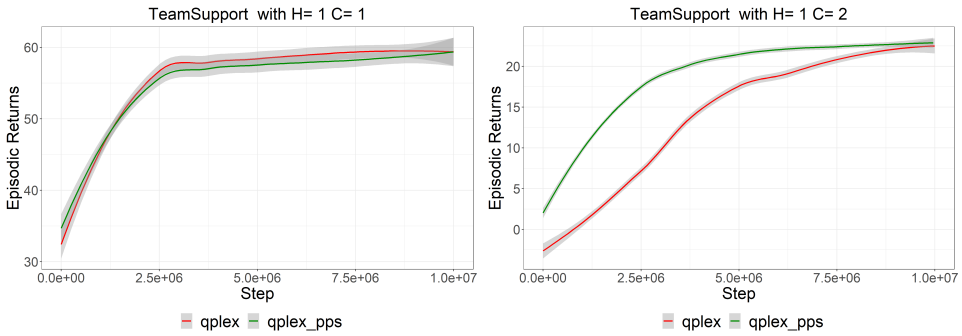
In this section, we present the necessary implementation details for reproducing our results. We first present the algorithm's hyperparameters, next, we present the architectures used for each algorithm, finally, we present the environments hyperparameters.

ALGORITHMS HYPERPARAMETERS

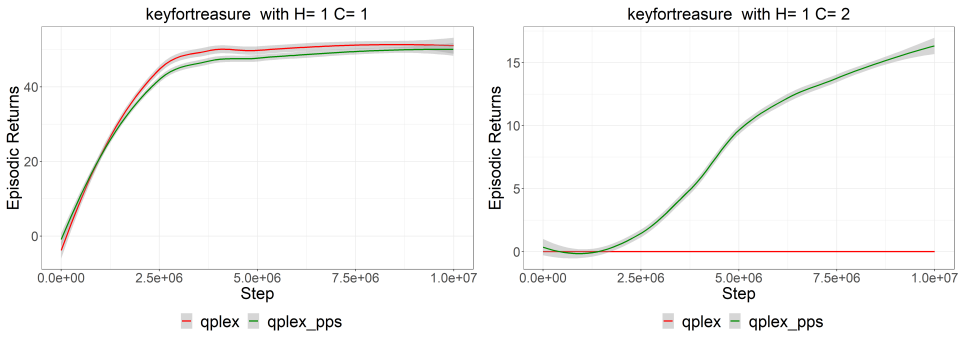
In this section, we present the relevant hyperparameters related to training the algorithms showcased in our paper. The hyperparameters shown in this section are kept fixed throughout all three environments. MAPPO, IPPO and SAF are all trained using Proximal Policy Optimization and Table 6.3 summarizes the training hyperparameters for each algorithm. IPPO and MAPPO are trained with generalised Advantage Estimation (GAE) while SAF is not. All algorithms are trained with Adam optimiser with a fixed learning rate of 0.0007 throughout training, a weight decay of 0 and $\epsilon = 10^{-5}$.



(a)



(b)



(c)

Figure 6.8: Performance comparison between QPLEX baseline and QPLEX with pool of policies in different environments with 5 agents. The reason only pool of policies but not share knowledge source was used is because QPLEX already has a similar mechanism. The results suggest that a shared pool of policies among agents improve learning efficiency when coordination level is high

	SAF	MAPPO	IPPO
Learning rate	0.0007	0.0007	0.0007
PPO update epochs	10	10	10
Number of minibatches	1	1	1
Discount rate γ	0.99	0.99	0.99
GAE	No	Yes	Yes
GAE's λ	-	0.95	0.95
Entropy loss coefficient	0.01	0.01	0.01
Value loss coefficient	0.5	0.5	0.5
Advantage Normalization	Yes	Yes	Yes
Value loss clipping value	0.2	0.2	0.2
Gradient norm clipping value	9	10	10
Value loss coefficient	0.5	0.5	0.5
optimiser	Adam	Adam	Adam
optimiser's epsilon (ϵ)	1e-5	1e-5	1e-5
Weight decay	0	0	0

Table 6.3: Hyperparameters used for training the MARL algorithms across all the HECOGrid environments.

ARCHITECTURAL HYPERPARAMETERS

In this section, we present the exact architectures along with the hyperparameters that were used for each algorithm.

Actor and Critic Architectures Listing 6.1 illustrates the pytorch-style implementations of the actor and critic architectures. While MAPPO shares parameters for both the actor and the critic, IPPO trains separate networks for both the actor and the critic. SAF uses the same architecture (and the same hyperparameters) as IPPO and MAPPO for the actor network, with the difference that SAF initializes a pool of policies for each agent. Listing 6.2 shows a pytorch-style implementation of a CNN that acts as our feature extractor. For IPPO and MAPPO, each agent's observation is fed to the CNN to generate a feature vector $z \in \mathbb{R}^{N \times C}$ where $N = 10$ is the number of agents and $C = 64$ the hidden dimension. The feature vector z is fed to the actor network to get the action probabilities. For IPPO, the feature vector z is fed as is to the critic to get the value function, while for MAPPO, a vector $\tilde{z} = \texttt{concatenate}(z, \texttt{dim=-1}) \in \mathbb{R}^{NC}$ is formed by concatenating feature vectors from all agents and then fed to the critic, that's why we make the distinction between different critic architectures in Listing 6.1.

```
from torch import nn


n_agents = 10

def layer_init(layer, std=np.sqrt(2), bias_const=0.0):
```

```
    nn.init.orthogonal_(layer.weight, std)
    nn.init.constant_(layer.bias, bias_const)
    return layer
# actor's output is a vector of 7 channels which corresponds to the number of actions.
actor = nn.Sequential(
    layer_init(nn.Linear(64, 128)),
    nn.Tanh(),
    layer_init(nn.Linear(128, 128)),
    nn.Tanh(),
    layer_init(nn.Linear(128, 7))
)

critic_ippo = nn.Sequential(
    layer_init(nn.Linear(64, 128)),
    nn.Tanh(),
    layer_init(nn.Linear(128, 128)),
    nn.Tanh(),
    layer_init(nn.Linear(128, 1))
)

critic_mappo = nn.Sequential(
    layer_init(nn.Linear(64 * n_agents, 128)),
    nn.Tanh(),
    layer_init(nn.Linear(128, 128)),
    nn.Tanh(),
    layer_init(nn.Linear(128, 1))
)
```

Listing 6.1: Pytorch-style implementation of the actor and critic architectures with the hyperparameters used in the paper for IPPO and MAPPO. We also provide implementation for the orthogonal initialization scheme.

```
from torch import nn
import torch.nn.functional as F

class CNN(nn.Module):
    def __init__(
        self,
        in_channels=3,
        channels=[32, 64],
        kernel_sizes=[4, 3],
        strides=[2, 2],
        hidden_layer=512,
        out_size=64):

        super().__init__()

        self.conv1 = nn.Conv2d(in_channels, channels[0], kernel_sizes[0], strides[0])
        self.conv2 = nn.Conv2d(channels[0], channels[1], kernel_sizes[1], strides[1])
        self.linear1 = nn.Linear(2304, hidden_layer)
        self.linear2 = nn.Linear(hidden_layer, out_size)

    def forward(self, inputs):
        x = F.relu(self.conv1(inputs / 255.))
```

```
        x = F.relu(self.conv2(x))
        x = x.reshape(x.shape[0], -1)
        x = F.relu(self.linear1(x))
        x = self.linear2(x)

        return x
```

Listing 6.2: Pytorch-style implementation of the CNN that generates a feature vector from the observations. The feature vector is then input to the actor and the critic.

Mapping function Architectures SAF makes use of MLPs architectures to encode the partial observations, messages, policies and to combine encoded state and encoded messages. Each agent's observation is encoded with a g_θ that is a CNN (see Listing 6.2) which generates a feature vector z. The agent's states are derived from the observation encoding using g_w which is the State Projector. g_ϕ projects the concatenation of the agent's state and message. Finally, g_{psel} is an MLP that encodes z and is used to select a policy from the pool of policies. See Table 6.4 for an overview of the MLP architectures and their hyperparameters.

g_θ	g_w	g_{psel}	g_ϕ
FC(64)	FC(64)	FC(64)	FC(64)
FC(128)	FC(128)	FC(128)	FC(128)
Input State [64]	Input State [64]	Input State [64]	Input State [128]

Table 6.4: Mapping functions used to encode partial observations, messages, policies and to combine encoded state and encoded messages.

Knowledge Source Hyperparameters We present the hyperparameters used in the `Perceiver-IO` architecture [323] that makes up the knowledge source. We use 2 Perceiver layers and use $L = 4$ slots for the knowledge source. The number of policies in the pool is set to 4. Table 6.5 summarizes the hyperparameters used to define both the Perceiver Encoder and the Cross Attention Layer.

ENVIRONMENT HYPERPARAMETERS

This section presents the hyperparameters for our three environments excluding coordination/heterogeneity levels since those are experiment dependent and are clarified in the main text. The agents

`Perceiver-IO` **Hyperparameters**	Values
PerceiverEncoder Hyperparameters	
latent dimension	4
num latent channels	64
cross attention channels	64
self attention heads	1
self attention layers per block	1
self attention blocks	1
dropout	No
CrossAttention Hyperparameters	
query dimension	64
key, value dimension	64
num query, key channels	64
num value channels	64
dropout	No

Table 6.5: Hyperparameters used to define the `Perceiver-IO` architecture used within the Knowledge Source.

	`TeamTogether`	`TeamSupport`	`KeyForTreasure`
Gym Observation Space	`Box(0,255,(28,28,3),dtype=uint8)`		
Gym Action Space	`Discrete(7)`		
Number of treasures	100	100	100
Grid size	30	30	30
Max. number of steps/episode	50	50	50
Partial View Size	7	7	7
View Tile Size	4	4	4
Clutter Density	0.1	0.1	0.1

Table 6.6: The partial view size parameter controls how much of the grid the agent can see.

7

CONCLUSION

In this dissertation, I explore how constructing autoregressive deep state-space models for video prediction and world modeling advance the development of artificial intelligence systems capable of predicting future states, adapting to new scenarios, and making reliable decisions with finite data. The research centers around two closely related objectives: forecasting future frames in video prediction tasks (i.e., Chapters 1 to 4), and addressing the broader challenge of modeling environmental dynamics through learning the causal relationship between actions and latent representations (i.e., Chapters 5 and 6).

Specifically, I investigate how different inductive biases (e.g., disentanglement, temporal consistency, and masked generative priors) influence the learning behaviour of generative models within video prediction and world modeling frameworks. While the proposed methods demonstrate performance gains across various experimental setups (e.g., precipitation forecasting, and games), the true value of this dissertation lies in the versatility of the proposed inductive biases. Being developed and evaluated in diverse contexts, the proposed inductive biases are designed to be applied to large-scale architectures such as large language models (LLMs) as well. Ultimately, the same algorithmic principles can be adapted for vision-driven control tasks or anticipating rare climate events with potentially significant societal impacts.

The main contributions within this dissertation are as follows:

- Chapter 1: We introduce α-TCVAE, a convex lower bound on the joint total correlation between input and learned latent representations, generalising β-VAE while combining the Variational Information Bottleneck [91] and Conditional Entropy Bottleneck principles [109]. The proposed approach achieves superior representation and generative capabilities, excelling in benchmarks like MPI3D-Real and model-based reinforcement learning tasks.

- Chapter 2: We present Conditional Autoregressive Slot Attention, a framework to improve the temporal consistency of extracted object-centring representations. In this work we show how temporal consistency improves the performance on video-based downstream tasks.

- Chapter 3: We propose NowcastingGPT, a video prediction model optimised with an Extreme Value Loss to better represent and predict extreme precipitation events. Moreover, through the use of a classifier, the proposed approach classifies tokens related to extreme events. As a result, the proposed approach outperforms all the related baselines and sets a new state-of-the-art benchmark on the KNMI dataset, which contains precipitation maps of the Netherlands from 2008 to 2024.

- Chapter 4: In this chapter we present a physics-informed variation of NowcastingGPT, which integrates physical constraints (e.g., Moisture Conservation), and is able to perform consistent predictions, outperforming all considered baselines.

- Chapter 5: We propose GIT-STORM, a novel world model that uses a masked generative prior in its dynamics module. The proposed approach presents state-of-the-art performances on the Atari100k benchmark, while showing significant improvements on the DeepMind Control Suite benchmark. Moreover, in this work we extend transformer-based world models to continuous action environments (e.g., DMC Suite).

- Chapter 6: We introduce HECOGrid, a RL environment that allows to explicitly assess coordination among N agents for different environmental heterogeneity levels. Moreover, we propose Stateful Active Facilitator, a novel approach to improve agents' communication and coordination capabilities.

7.1. Future Research Directions

A fruitful path for subsequent studies is the investigation of hierarchical representations capable of capturing relationships at varying levels of abstraction. Such an approach would incorporate both short-term dynamics and long-term strategies, enabling agents to reuse previously acquired skills and interpret new environments. Another valuable direction would be to refine objectives for learning world models, moving away from pure reconstruction-based criteria to more semantically grounded signals. This could include object-centric representation learning and the integration of language-based priors, which would allow agents to have better reasoning and generalization capabilities.

Another interesting line of work could be using efficient parameter fine-tuning approaches (e.g., LoRA [342]) to turn large pretrained models, such as multimodal large language-vision models (e.g., Chameleon [343]), into a world model. The idea is to retrain only the dynamics head and use LoRA blocks to learn the state-transition dynamics of the considered dataset. As a result, rather than retraining the whole world model pipeline from scratch, we could train only the missing modules (e.g., dynamics head) and rely on pretrained models for the remaining ones. Such an approach, not only should present higher generalization capabilities, but would also have a much smaller carbon footprint than world models trained completely from scratch.

A compelling future research direction lies in explicitly integrating causal reasoning and counterfactual inference into world models. Current world models primarily focus on predicting what will happen, but understanding why things happen and what would have happened under different circumstances is crucial for robust decision-making and adaptation, especially in complex and uncertain environments. This could involve exploring techniques from causal representation learning [344, 345] to discover the underlying causal structure of the environment from observational data. Furthermore, incorporating counterfactual reasoning mechanisms, inspired by works like [346, 347], would allow agents to evaluate the consequences of different actions in their imagined rollouts, leading to more informed and strategic planning. For instance, in a robotic navigation task, a causal world model could reason that bumping into an obstacle causes the robot to stop, and a counterfactual query could help determine what would have happened if the robot had taken a slightly different path. This direction could leverage recent advancements in differentiable causal discovery [348] and the development of causal-aware latent variable models [349], potentially leading to world models that are not only predictive but also capable of sophisticated reasoning about interventions and their effects.

An exciting and potentially transformative future direction involves developing a world model for code generation and understanding. The application of world models to domains beyond traditional physical environments, such as code compilation and software development, has demonstrated enormous potential [350]. Designing a world model where a generative model can understand the algorithmic underlying structure and predict subsequent snippets of code, whilst an agent can take actions in the form of generating code to solve specific programming tasks, presents a significant research opportunity. This could involve training models on large datasets of code to learn the syntax, semantics, and common patterns, enabling them to anticipate the next lines of code or to identify potential errors. Furthermore, an agent equipped with such a world model could perform in-silico "rollouts" of different code generation strategies to debug and refine its approach before executing the code in a real environment. This direction could build upon recent advancements in large language models for code [351, 352], exploring how to structure them within a world model framework to enable more robust and efficient code generation and understanding.

7.2. BROADER IMPACT

From a societal perspective, the innovations presented in this work have the potential to expand the accessibility and sustainability of AI across multiple domains. More efficient and generalizable learning strategies reduce the costs associated with data collection and heavy compute, enabling smaller institutions and a broader array of industries to adopt and improve upon state-of-the-art methods. By making AI more inclusive, we can unlock new applications in healthcare, assistive technology, manufacturing, and education, all of which benefit from robust, low-data, and adaptively deployable predictive models. Yet these advances may also accelerate the pace of automation, demanding attention to policy, ethics, and workforce transitions as intelligent systems grow increasingly capable.

In summary, this doctoral dissertation has demonstrated that predictive models—founded on disentangled, hierarchically structured, and computationally efficient approaches—can improve data efficiency and decision-making across a spectrum of tasks. These insights set the groundwork for future research into universal world models that span across domains, incorporate diverse prior knowledge, and function reliably in complex real-world scenarios. Ultimately, the convergence of generative modeling, reinforcement learning, multi-agent systems, and domain-specific constraints underscores the transformative role of predictive intelligence in shaping the next generation of AI systems.

BIBLIOGRAPHY

[1] E. C. Tolman. "Cognitive maps in rats and men". In: *Psychological Review* 55.4 (1948), pp. 189–208.

[2] Y. LeCun. *Deep Learning of Representations for Unsupervised and Transfer Learning*. 2015. arXiv: 1505.06411.

[3] A. M. Turing. "Computing Machinery and Intelligence". In: *Mind* LIX.236 (1950), pp. 433–60.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2012, pp. 1097–1105.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[6] L. P. Kaelbling, M. L. Littman, and A. W. Moore. "Reinforcement learning: A survey". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285.

[7] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. "Benchmarking Deep Reinforcement Learning for Continuous Control". In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1329–1338.

[8] D. Ha and J. Schmidhuber. "Recurrent World Models Facilitate Policy Evolution". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018, pp. 2450–2462.

[9] L. Castrejón, N. Ballas, and A. Courville. "Improved Conditional VRNNs for Video Prediction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 7608–7617.

[10] R. Mottaghi, M. Rastegari, J. Redmon, D. Ross, S. Fidler, R. Urtasun, and A. Farhadi. "Newtonian Image Understanding: Unfolding the Dynamics of Objects in Static Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3521–3529.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Convolutional Networks for Images, Speech, and Time-series". In: *The Handbook of Brain Theory and Neural Networks*. 1995, pp. 255–258.

[12] S. Hochreiter and J. Schmidhuber. "Long Short-term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[13] D. Ha and J. Schmidhuber. "Recurrent World Models Facilitate Policy Evolution". In: *Neural Information Processing Systems (NeurIPS)*. 2018.

[14] D. P. Kingma and M. Welling. “Auto-encoding Variational Bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[16] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. “A Recurrent Latent Variable Model for Sequential Data”. In: *Neural Information Processing Systems (NeurIPS)*. 2015.

[17] J. Ho, A. Jain, and P. Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[18] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. “Score-Based Generative Modeling through Stochastic Differential Equations”. In: *International Conference on Learning Representations*. 2021.

[19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[20] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[21] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. “Dream to Control: Learning Behaviors by Latent Imagination”. In: *International Conference on Learning Representations (ICLR)*. 2020.

[22] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. “DayDreamer: World Models for Physical Robot Learning”. In: *Conference on Robot Learning (CoRL)*. 2023.

[23] R. Sakagami, F. S. Lay, A. Dömel, M. J. Schuster, A. Albu-Schäffer, and F. Stulp. “Robotic world models—conceptualization, review, and engineering best practices”. In: *Frontiers in Robotics and AI* 10 (2023), p. 1253049.

[24] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. “Learning Latent Dynamics for Planning from Pixels”. In: *International Conference on Machine Learning (ICML)*. 2019.

[25] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. “Mastering Atari with Discrete World Models”. In: *International Conference on Learning Representations (ICLR)*. 2021.

[26] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. “Mastering diverse domains through world models”. In: *arXiv preprint arXiv:2301.04104* (2023).

[27] M. Okada and T. Taniguchi. “Dreaming: Model-based Reinforcement Learning by Latent Imagination without Reconstruction”. In: *International Conference on Robotics and Automation (ICRA)*. 2021.

[28] M. Okada and T. Taniguchi. "DreamingV2: Reinforcement Learning with Discrete World Models without Reconstruction". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2022.

[29] F. Deng, I. Jang, and S. Ahn. "DreamerPro: Reconstruction-Free Model-Based Reinforcement Learning with Prototypical Representations". In: *International Conference on Machine Learning (ICML)*. 2022.

[30] J. Ha, K. Kim, and Y. Kim. "Dream to generalise: Zero-shot model-based reinforcement learning for unseen visual distractions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6. 2023, pp. 7802–7810.

[31] Y. LeCun and Courant. *A Path Towards Autonomous Machine Intelligence Version 0.9.2, 2022-06-27*. 2022. URL: https://api.semanticscholar.org/CorpusID:251881108.

[32] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas. "Self-Supervised Learning From Images With a Joint-Embedding Predictive Architecture". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[33] Z. Fei, M. Fan, and J. Huang. *A-JEPA: Joint-Embedding Predictive Architecture Can Listen*. 2024. arXiv: 2311.15830.

[34] A. Bardes, J. Ponce, and Y. LeCun. *MC-JEPA: A Joint-Embedding Predictive Architecture for Self-Supervised Learning of Motion and Content Features*. 2023. arXiv: 2307.12698.

[35] A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas. *Revisiting Feature Prediction for Learning Visual Representations from Video*. 2024. arXiv: 2404.08471.

[36] C. Chen, Y.-F. Wu, J. Yoon, and S. Ahn. *TransDreamer: Reinforcement Learning with Transformer World Models*. 2022. arXiv: 2202.09481.

[37] V. Micheli, E. Alonso, and F. Fleuret. "Transformers are Sample-Efficient World Models". In: *International Conference on Learning Representations (ICLR)*. 2023.

[38] J. Robine, M. Höftmann, T. Uelwer, and S. Harmeling. "Transformer-based World Models Are Happy With 100k Interactions". In: *International Conference on Learning Representations (ICLR)*. 2023.

[39] W. Zhang, G. Wang, J. Sun, Y. Yuan, and G. Huang. "STORM: Efficient Stochastic Transformer based World Models for Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 27147–27166. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/5647763d4245b23e6a1cb0a8947b38c9-Paper-Conference.pdf.

[40] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, ..., N. de Freitas, S. Singh, and T. Rocktäschel. *Genie: Generative Interactive Environments*. 2024. arXiv: 2402.15391.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.* "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), p. 529.

[42] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. "Rainbow: Combining improvements in deep reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 2018.

[43] M. Laskin, A. Srinivas, and P. Abbeel. "Curl: Contrastive unsupervised representations for reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5639–5650.

[44] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, *et al.* "Model Based Reinforcement Learning for Atari". In: *International Conference on Learning Representations (ICLR)*. 2020.

[45] Y. Gal, R. McAllister, and C. E. Rasmussen. *Improving PILCO with Bayesian Neural Network Dynamics Models*. 2016. arXiv: 1605.07129.

[46] M. Henaff, A. Canziani, and Y. LeCun. "Model-Predictive Policy Learning with Uncertainty Regularization for Driving in Dense Traffic". In: *International Conference on Robotics and Automation (ICRA)*. 2018.

[47] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. M. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. "DeepMind Control Suite". In: *arXiv preprint arXiv:1801.00690*. 2018.

[48] W. Huang, J. Ji, B. Zhang, C. Xia, and Y. Yang. "SafeDreamer: Safe Reinforcement Learning with World Models". In: *International Conference on Learning Representations (ICLR)*. 2024.

[49] H. Ma, J. Wu, N. Feng, C. Xiao, D. Li, J. Hao, J. Wang, and M. Long. *HarmonyDream: Task Harmonization Inside World Models*. 2024. arXiv: 2310.00344.

[50] S. Zhou, Y. Du, J. Chen, Y. Li, D.-Y. Yeung, and C. Gan. *RoboDreamer: Learning Compositional World Models for Robot Imagination*. 2024. arXiv: 2404.12377 [cs.RO].

[51] S. Yang, Y. Du, S. K. Seyed Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel. "Learning Interactive Real-World Simulators". In: *International Conference on Learning Representations (ICLR)*. 2024.

[52] A. Piergiovanni, M. S. Ryoo, and A. Angelova. *Learning Composable Models of Robot Skills with Differentiable Symbolic Planning*. 2019. arXiv: 1905.12197.

[53] J. Y. Koh, H. Lee, Y. Yang, J. Baldridge, and P. Anderson. "Pathdreamer: A World Model for Indoor Navigation". In: 2021.

[54] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. "Planning to Explore via Self-Supervised World Models". In: *International Conference on Machine Learning (ICML)*. 2020.

[55] R. Mendonca, S. Bahl, and D. Pathak. *Structured World Models from Human Videos*. 2023. arXiv: 2308.10901 [cs.RO]. URL: https://arxiv.org/abs/2308.10901.

[56] B. J. Baars. *A Cognitive Theory of Consciousness*. New York: Cambridge University Press, 1988.

[57] A. V. D. Oord, O. Vinyals, and et al. "Neural discrete representation learning". In: *Advances in Neural Information Processing Systems* 30 (2017).

[58] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado. "Gaia-1: A generative world model for autonomous driving". In: *arXiv preprint arXiv:2309.17080* (2023).

[59] H. Bi, M. Kyryliuk, Z. Wang, C. Meo, Y. Wang, R. Imhoff, R. Uijlenhoet, and J. Dauwels. "Nowcasting of Extreme Precipitation Using Deep Generative Models". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10094988.

[60] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. "Videogpt: Video generation using vq-vae and transformers". In: *arXiv preprint arXiv:2104.10157* (2021).

[61] A. Kumar, T. Islam, Y. Sekimoto, C. Mattmann, and B. Wilson. "Convcast: An embedded convolutional LSTM based architecture for precipitation nowcasting using satellite data". In: *Plos one* 15.3 (2020), e0230114.

[62] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. "A recurrent latent variable model for sequential data". In: *Advances in neural information processing systems*. 2015, pp. 2980–2988.

[63] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. "Video pretraining (vpt): Learning to act by watching unlabelled online videos". In: *arXiv preprint arXiv:2206.11795* (2022).

[64] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa, *et al.* "Magvit: Masked generative video transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 10459–10469.

[65] P. Esser, R. Rombach, and B. Ommer. "Taming transformers for high-resolution image synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12873–12883.

[66] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. "Object-centric learning with slot attention". In: *arXiv preprint arXiv:2006.15055* (2020).

[67] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner. "Spatial broadcast decoder: A simple architecture for learning disentangled representations in VAEs". In: *arXiv preprint ArXiv:1901.07017* (2019).

[68] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.* "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[69] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.* "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021.

[70] D. Mizrahi, R. Bachmann, O. Kar, T. Yeo, M. Gao, A. Dehghan, and A. Zamir. "4M: Massively Multimodal Masked Modeling". In: *Advances in Neural Information Processing Systems* 36 (2024).

[71] A. Trask, F. Hill, S. E. Reed, J. Rae, C. Dyer, and P. Blunsom. "Neural arithmetic logic units". In: *Advances in Neural Information Processing Systems*. 2018, pp. 8035–8044.

[72] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, *et al.* "Dota 2 with large scale deep reinforcement learning". In: *arXiv preprint arXiv:1912.06680* (2019).

[73] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019.

[74] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[75] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.* "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[76] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[77] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[78] A. Goyal, A. Didolkar, N. R. Ke, C. Blundell, P. Beaudoin, N. Heess, M. C. Mozer, and Y. Bengio. "Neural production systems". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25673–25687.

[79] J. Yin, C. Meo, A. Roy, Z. B. Cher, M. Lică, Y. Wang, R. Imhoff, R. Uijlenhoet, and J. Dauwels. "Precipitation Nowcasting Using Physics Informed Discriminator Generative Models". In: *2024 32nd European Signal Processing Conference (EUSIPCO)*. 2024, pp. 967–971. DOI: 10.23919/EUSIPCO63174.2024.10715141.

[80] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. Vol. 135. MIT Press Cambridge, 1998.

[81] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. "Mastering Atari with discrete world models". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=0oabwyZbOu.

[82] Ł. Kaiser, M. Babaeizadeh, P. Miłos, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski. "Model based reinforcement learning for atari". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=S1xCPJHtDB.

[83] V. Micheli, E. Alonso, and F. Fleuret. "Transformers are Sample-Efficient World Models". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=vhFu1Acb0xb.

[84] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. "Dream to control: Learning behaviors by latent imagination". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=S1lOTC4tDS.

[85] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. "Mastering Atari games with limited data". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25476–25488.

[86] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, *et al.* "Mastering Atari, Go, chess and Shogi by planning with a learned model". In: *Nature* 588.7839 (2020), pp. 604–609.

[87] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551.

[88] Y. Bengio, N. Léonard, and A. Courville. "Estimating or propagating gradients through stochastic neurons for conditional computation". In: *arXiv preprint arXiv:1308.3432* (2013).

[89] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.* "Improving language understanding by generative pre-training". In: (2018).

[90] Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828.

[91] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. "Deep variational information bottleneck". In: *International Conference on Learning Representations* (2017).

[92] Y. Li. "Deep reinforcement learning: An overview". In: *arXiv preprint ArXiv:1701.07274* (2017).

[93] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76.

[94] X. Sun, J. Gu, and H. Sun. "Research progress of zero-shot learning". In: *Applied Intelligence* 51.6 (2021), pp. 3600–3614.

[95] A. Radford, L. Metz, and S. Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint ArXiv:1511.06434* (2016).

[96] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4114–4124.

[97] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. "beta-vae: Learning basic visual concepts with a constrained variational framework". In: (2016).

[98] H. Kim and A. Mnih. "Disentangling by factorising". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2649–2658.

[99] A. Kirsch, C. Lyle, and Y. Gal. *Unpacking Information Bottlenecks: Surrogate Objectives for Deep Learning*. 2021. URL: https://openreview.net/forum?id=5rc0K0ezhqI.

[100] K. Do and T. Tran. "Theory and Evaluation Metrics for Learning Disentangled Representations". In: *International Conference on Learning Representations*. 2020.

[101] D. Friedman and A. B. Dieng. "The vendi score: A diversity evaluation metric for machine learning". In: *arXiv preprint ArXiv:2210.02410* (2022).

[102] R. Child. "Very deep vaes generalise autoregressive models and can outperform them on images". In: *arXiv preprint arXiv:2011.10650* (2020).

[103] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. "Towards a definition of disentangled representations". In: *Theoretical Physics for Deep Learning Workshop, ICML*. 2019.

[104] I. Higgins, S. Racanière, and D. Rezende. "Symmetry-based representations for artificial and biological general intelligence". In: *Frontiers in Computational Neuroscience* 16 (2022), p. 836498.

[105] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. DOI: 10.1017/CBO9780511804090.

[106] R. T. Chen, X. Li, R. Grosse, and D. Duvenaud. "Isolating sources of disentanglement in variational autoencoders". In: *arXiv preprint arXiv:1802.04942* (2018).

[107] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. "Disentangling disentanglement in variational autoencoders". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4402–4412.

[108] M. Rolinek, D. Zietlow, and G. Martius. "Variational autoencoders pursue pca directions (by accident)". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12406–12415.

[109] I. Fischer and A. A. Alemi. "CEB improves model robustness". In: *Entropy* 22.10 (2020), p. 1081.

[110] M. W. Gondal, M. Wuthrich, D. Miladinovic, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer. "On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[111] D. Hafner, K.-H. Lee, I. Fischer, and P. Abbeel. "Deep Hierarchical Planning from Pixels". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 26091–26104.

[112] C. Burgess and H. Kim. *3D Shapes Dataset*. https://github.com/deepmind/3dshapes-dataset/. 2018.

[113] W. Chen, H. Xu, Z. Li, D. Pei, J. Chen, H. Qiao, Y. Feng, and Z. Wang. "Unsupervised anomaly detection for intricate KPIs via adversarial training of VAE". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE. 2019, pp. 1891–1899.

[114] Y. Shi, B. Paige, P. Torr, *et al.* "Variational mixture-of-experts autoencoders for multi-modal deep generative models". In: *Advances in Neural Information Processing Systems* 32 (2019).

[115] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. Wu. "Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and autoregressive prosody prior". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 6699–6703.

[116] Y. Li, C. Yu, G. Sun, W. Zu, Z. Tian, Y. Wen, W. Pan, C. Zhang, J. Wang, Y. Yang, *et al.* "Cross-Utterance Conditioned VAE for Speech Generation". In: *arXiv preprint ArXiv:2309.04156* (2023).

[117] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia. "Pianotree VAE: Structured representation learning for polyphonic music". In: *arXiv preprint ArXiv:2008.07118* (2020).

[118] J. Peng, D. Liu, S. Xu, and H. Li. "Generating diverse structure for image inpainting with hierarchical vq-VAE". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10775–10784.

[119] A. Vahdat and J. Kautz. "NVAE: A deep hierarchical variational autoencoder". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 19667–19679.

[120] A. Razavi, A. v. d. Oord, and O. Vinyals. "Generating diverse high-fidelity images with vq-vae-2". In: *arXiv preprint arXiv:1906.00446* (2019).

[121] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. "Understanding disentangling in β-VAE". In: *arXiv preprint ArXiv:1804.03599* (2018).

[122] K. Roth, M. Ibrahim, Z. Akata, P. Vincent, and D. Bouchacourt. "Disentanglement of Correlated Factors via Hausdorff Factorized Support". In: *International Conference on Learning Representations (ICLR)*. 2023.

[123] R. Shwartz-Ziv and N. Tishby. "Opening the black box of deep neural networks via information". In: *arXiv preprint ArXiv:1703.00810* (2017).

[124] N. Tishby, F. C. Pereira, and W. Bialek. "The information bottleneck method". In: *Proceedings of the 37th Allerton Conference on Communication, Control and Computation* (2001).

[125] H. Hwang, G.-H. Kim, S. Hong, and K.-E. Kim. "Multi-View Representation Learning via Total Correlation Objective". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. 2021.

[126] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang. "Mode seeking generative adversarial networks for diverse image synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1429–1437.

[127] Z. Lin, T. Sercu, Y. LeCun, and A. Rives. "Deep generative models create new and diverse protein structures". In: *Machine Learning for Structural Biology Workshop, NeurIPS*. 2021.

[128] Z. Wu, K. E. Johnston, F. H. Arnold, and K. K. Yang. "Protein sequence design with deep generative models". In: *Current Opinion in Chemical Biology* 65 (2021), pp. 18–27.

[129] R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, and A. Celikyilmaz. "How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven". In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 652–670.

[130] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. "Diverse image-to-image translation via disentangled representations". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 35–51.

[131] H. Kazemi, S. M. Iranmanesh, and N. Nasrabadi. "Style and content disentanglement in generative adversarial networks". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 848–856.

[132] Y. Li, K. K. Singh, U. Ojha, and Y. J. Lee. "Mixnmatch: multifactor disentanglement and encoding for conditional image generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8039–8048.

[133] C. Eastwood and C. K. Williams. "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations*. 2018.

[134] S. Gao, R. Brekelmans, G. Ver Steeg, and A. Galstyan. "Auto-encoding total correlation explanation". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1157–1166.

[135] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in Neural Information Processing Systems* 30 (2017).

[136] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem. "On the fairness of disentangled representations". In: *Advances in Neural Information Processing Systems* 32 (2019).

[137] P. Moreno, C. K. Williams, C. Nash, and P. Kohli. "Overcoming occlusion with inverse graphics". In: *European Conference on Computer Vision*. Springer. 2016, pp. 170–185.

[138] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. "Deep Visual Analogy-Making". In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015.

[139] Z. Liu, P. Luo, X. Wang, and X. Tang. "Deep learning face attributes in the wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[140] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 2017.

[141] L. Mahon, L. Shah, and T. Lukasiewicz. "Correcting Flaws in Common Disentanglement Metrics". In: *arXiv preprint ArXiv:2304.02335* (2023).

[142] J. Cao, R. Nai, Q. Yang, J. Huang, and Y. Gao. "An Empirical Study on Disentanglement of Negative-free Contrastive Learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 1210–1222.

[143] T. Karras, S. Laine, and T. Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[144] C. Chadebec, L. Vincent, and S. Allassonnière. "Pythae: Unifying Generative Autoencoders in Python-A Benchmarking Use Case". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21575–21589.

[145] A. Zadaianchuk, M. Seitzer, and G. Martius. "Object-centric learning for real-world videos by predicting temporal feature similarities". In: *Advances in Neural Information Processing Systems* 36 (2023).

[146] Z. Wu, J. Hu, W. Lu, I. Gilitschenski, and A. Garg. "Slotdiffusion: Object-centric generative modeling with diffusion models". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 50932–50958.

[147] I. Kakogeorgiou, S. Gidaris, K. Karantzalos, and N. Komodakis. *SPOT: Self-Training with Patch-Order Permutation for Object-Centric Learning with Autoregressive Transformers*. 2023.

[148] G. Singh, F. Deng, and S. Ahn. "Illiterate DALL-E Learns to Compose". In: *International Conference on Learning Representations*. 2022.

[149] M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C.-J. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, *et al.* "Bridging the gap to real-world object-centric learning". In: *International Conference on Learning Representations*. 2023.

[150] D. Erhan, A. Courville, and Y. Bengio. "Understanding representations learned in deep architectures". In: *Department dInformatique et Recherche Operationnelle, University of Montreal, QC, Canada, Tech. Rep* 1355.1 (2010), p. 69.

[151] Y. Bengio. "Deep learning of representations for unsupervised and transfer learning". In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 17–36.

[152] X. Glorot, A. Bordes, and Y. Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach". In: *ICML*. 2011.

[153] A. A. Duval, V. Schmidt, A. Hernández-García, S. Miret, F. D. Malliaros, Y. Bengio, and D. Rolnick. "FAENet: Frame Averaging Equivariant GNN for Materials Modeling". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. PMLR, 2023, pp. 9013–9033.

[154] A. Goyal, A. Lamb, P. Gampa, P. Beaudoin, C. Blundell, S. Levine, Y. Bengio, and M. C. Mozer. "Factorizing Declarative and Procedural Knowledge in Structured, Dynamical Environments". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=VVdmjgu7pKM.

[155] A. Goyal, A. Didolkar, N. R. Ke, C. Blundell, P. Beaudoin, N. Heess, M. Mozer, and Y. Bengio. "Neural Production Systems". In: *arXiv preprint arXiv:2103.01937* (2021).

[156] G. Singh, Y.-F. Wu, and S. Ahn. "Simple unsupervised object-centric learning for complex and naturalistic videos". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18181–18196.

[157] T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, and K. Greff. "Conditional Object-Centric Learning from Video". In: *International Conference on Learning Representations*. 2022.

[158] Z. Wu, N. Dvornik, K. Greff, T. Kipf, and A. Garg. "SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models". In: *The Eleventh International Conference on Learning Representations*. 2023.

[159] A. Goyal, A. Lamb, P. Gampa, P. Beaudoin, S. Levine, C. Blundell, Y. Bengio, and M. Mozer. "Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems". In: *arXiv preprint arXiv:2006.16225* (2020).

[160] K. Yi*, C. Gan*, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. "CLEVRER: Collision Events for Video Representation and Reasoning". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HkxYzANYDB.

[161] D. M. Bear, E. Wang, D. Mrowca, F. J. Binder, H.-Y. F. Tung, R. Pramod, C. Holdaway, S. Tao, K. Smith, L. Fei-Fei, *et al.* "Physion: Evaluating Physical Prediction from Vision in Humans and Machines". In: *arXiv preprint arXiv:2106.08261* (2021).

[162] Y. Bengio and J. Bergstra. "Slow, decorrelated features for pretraining complex cell-like networks". In: *Advances in neural information processing systems* 22 (2009).

[163] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang. "Learning blind video temporal consistency". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 170–185.

[164] G. Eilertsen, R. K. Mantiuk, and J. Unger. "Single-frame regularization for temporally stable cnns". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11176–11185.

[165] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross. "Practical temporal consistency for image-based graphics applications". In: *ACM Transactions on Graphics (ToG)* 31.4 (2012), pp. 1–8.

[166] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister. "Blind video temporal consistency". In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), pp. 1–9.

[167] X. Dong, B. Bonev, Y. Zhu, and A. L. Yuille. "Region-based temporally consistent video post-processing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 714–722.

[168] C.-H. Yao, C.-Y. Chang, and S.-Y. Chien. "Occlusion-aware video temporal consistency". In: *Proceedings of the 25th ACM international conference on Multimedia*. 2017, pp. 777–785.

[169] J. L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: https://www.sciencedirect.com/science/article/pii/036402139090002E.

[170] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf. "Recurrent Independent Mechanisms". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=mLcmdlEUxy-.

[171] G. Elsayed, A. Mahendran, S. Van Steenkiste, K. Greff, M. C. Mozer, and T. Kipf. "Savi++: Towards end-to-end object-centric learning from real-world videos". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 28940–28954.

[172] A. Didolkar, A. Goyal, and Y. Bengio. "Cycle consistency driven object discovery". In: *International Conference on Learning Representations*. 2024.

[173] E. S. Spelke. "Where perceiving ends and thinking begins: The apprehension of objects in infancy". In: *Perceptual development in infancy*. Psychology Press, 2013, pp. 197–234.

[174] A. Chakravarthy, T. Nguyen, A. Goyal, Y. Bengio, and M. C. Mozer. "Spotlight Attention: Robust Object-Centric Learning With a Spatial Locality Prior". In: *arXiv preprint arXiv:2305.19550* (2023).

[175] A. Nakano, M. Suzuki, and Y. Matsuo. "Interaction-Based Disentanglement of Entities for Object-Centric World Models". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=JQc2VowqCzz.

[176] A. Dittadi, S. Papa, M. De Vita, B. Schölkopf, O. Winther, and F. Locatello. "Generalization and robustness implications in object-centric learning". In: *arXiv preprint ArXiv:2107.00637* (2021).

[177] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. "Multi-object representation learning with iterative variational inference". In: *arXiv preprint arXiv:1903.00450* (2019).

[178] Y.-F. Wu, M. Lee, and S. Ahn. "Neural Language of Thought Models". In: *International Conference on Learning Representations*. 2024.

[179] B. Jia, Y. Liu, and S. Huang. "Improving Object-centric Learning with Query Optimization". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=_-FN9mJsgg.

[180] T. Kipf, E. van der Pol, and M. Welling. "Contrastive learning of structured world models". In: *arXiv preprint arXiv:1911.12247* (2019).

[181] N. R. Ke, A. Didolkar, S. Mittal, A. G. ALIAS PARTH GOYAL, G. Lajoie, S. Bauer, D. Jimenez Rezende, M. Mozer, Y. Bengio, and C. Pal. "Systematic Evaluation of Causal Discovery in Visual Model Based Reinforcement Learning". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Vol. 1. 2021.

[182] R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. Tenenbaum, and S. Levine. "Entity Abstraction in Visual Model-Based Reinforcement Learning". In: *Proceedings of the Conference on Robot Learning*. Ed. by L. P. Kaelbling, D. Kragic, and K. Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, 30 Oct–01 Nov 2020, pp. 1439–1456.

[183] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2901–2910.

[184] Z. Lin, Y.-F. Wu, S. Peri, B. Fu, J. Jiang, and S. Ahn. "Improving generative imagination in object-centric world models". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6140–6149.

[185] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[186] A. Van Den Oord, O. Vinyals, *et al.* "Neural discrete representation learning". In: *Advances in neural information processing systems* 30 (2017).

[187] D. Ding, F. Hill, A. Santoro, M. Reynolds, and M. Botvinick. "Attention over learned object embeddings enables complex visual reasoning". In: *Advances in neural information processing systems* 34 (2021), pp. 9112–9124.

[188] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly. *FVD: A new Metric for Video Generation*. 2019. URL: https://openreview.net/forum?id=rylgEULtdN.

[189] L. Alfieri, B. Bisselink, F. Dottori, G. Naumann, A. P. J. D. Roo, P. Salamon, K. Wyser, and L. Feyen. "Global projections of river flood risk in a warmer world". In: *Earth's Future* 5 (2017). URL: https://api.semanticscholar.org/CorpusID:42772267.

[190] M. Martinkova and J. Kysely. "Overview of observed Clausius-Clapeyron scaling of extreme precipitation in midlatitudes". In: *Atmosphere* 11.8 (2020), p. 786.

[191] S. Klocek, H. Dong, M. Dixon, P. Kanengoni, N. Kazmi, P. Luferenko, Z. Lv, S. Sharma, J. A. Weyn, and S. Xiang. "MS-nowcasting: Operational Precipitation Nowcasting with Convolutional LSTMs at Microsoft Weather". In: *ArXiv* abs/2111.09954 (2021). URL: https://api.semanticscholar.org/CorpusID:244463010.

[192] G. S. Czibula, A. Mihai, A.-I. Albu, I. G. Czibula, S. Burcea, and A. Mezghani. "AutoNowP: An Approach Using Deep Autoencoders for Precipitation Nowcasting Based on Weather Radar Reflectivity Prediction". In: *Mathematics* (2021). URL: https://api.semanticscholar.org/CorpusID:238018677.

[193] I. Malkin Ondík, L. Ivica, P. Šišan, I. Martynovskyi, D. Šaur, and L. Gaál. "A Concept of Nowcasting of Convective Precipitation Using an X-band Radar for the Territory of the Zlín Region (Czech Republic)". In: *Computer Science On-line Conference*. Springer, 2022, pp. 499–514.

[194] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems* 28 (2015).

[195] K. Trebing, T. Stanczyk, and S. Mehrkanoon. "SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture". In: *Pattern Recognition Letters* 145 (2021), pp. 178–186.

[196] C. Luo, X. Zhao, Y. Sun, X. Li, and Y. Ye. "PredRANN: The spatiotemporal attention Convolution Recurrent Neural Network for precipitation nowcasting". In: *Knowl. Based Syst.* 239 (2021), p. 107900. URL: https://api.semanticscholar.org/CorpusID:245591327.

[197] J. Liu, L. Xu, and N. Chen. "A spatiotemporal deep learning model ST-LSTM-SA for hourly rainfall forecasting using radar echo images". In: *Journal of Hydrology* (2022). URL: https://api.semanticscholar.org/CorpusID:247602986.

[198] M. Veillette, S. Samsi, and C. J. Mattioli. "SEVIR : A Storm Event Imagery Dataset for Deep Learning Applications in Radar and Satellite Meteorology". In: *Neural Information Processing Systems*. 2020. URL: https://api.semanticscholar.org/CorpusID:227222587.

[199] Y. Yang and S. Mehrkanoon. "AA-TransUNet: Attention Augmented TransUNet For Nowcasting Tasks". In: *2022 International Joint Conference on Neural Networks (IJCNN)* (2022), pp. 01–08. URL: https://api.semanticscholar.org/CorpusID:246705912.

[200] R. Prudden, S. Adams, D. Kangin, N. Robinson, S. Ravuri, S. Mohamed, and A. Arribas. "A review of radar-based nowcasting of precipitation and applicable machine learning techniques". In: *arXiv preprint arXiv:2005.04988* (2020).

[201] S. Pulkkinen, D. Nerini, A. A. Pérez Hortal, C. Velasco-Forero, A. Seed, U. Germann, and L. Foresti. "Pysteps: An open-source Python library for probabilistic precipitation nowcasting (v1. 0)". In: *Geoscientific Model Development* 12.10 (2019), pp. 4185–4219.

[202] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, *et al.* "Skilful precipitation nowcasting using deep generative models of radar". In: *Nature* 597.7878 (2021), pp. 672–677.

[203] H. Bi, M. Kyryliuk, Z. Wang, C. Meo, Y. Wang, R. Imhoff, R. Uijlenhoet, and J. Dauwels. "Nowcasting of Extreme Precipitation Using Deep Generative Models". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10094988.

[204] C. Bai, F. Sun, J. Zhang, Y. Song, and S. Chen. "Rainformer: Features Extraction Balanced Network for Radar-Based Precipitation Nowcasting". In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5. URL: https://api.semanticscholar.org/CorpusID:248132089.

[205] Q. Jin, X. Zhang, X. Xiao, Y. Wang, S. Xiang, and C. Pan. "Preformer: Simple and Efficient Design for Precipitation Nowcasting With Transformers". In: *IEEE Geoscience and Remote Sensing Letters* 21 (2024), pp. 1–5. DOI: 10.1109/LGRS.2023.3325628.

[206] C. Wu, J. Liang, L. Ji, F. Yang, Y. Fang, D. Jiang, and N. Duan. "Nüwa: Visual synthesis pre-training for neural visual world creation". In: *European conference on computer vision*. Springer. 2022, pp. 720–736.

[207] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. "Videogpt: Video generation using vq-vae and transformers". In: *arXiv preprint arXiv:2104.10157* (2021).

[208] P. Esser, R. Rombach, and B. Ommer. "Taming transformers for high-resolution image synthesis". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12873–12883.

[209] W. Yan, D. Hafner, S. James, and P. Abbeel. *Temporally Consistent Transformers for Video Generation*. 2023. arXiv: 2210.02396 [cs.CV].

[210] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He. "Modeling Extreme Events in Time Series Prediction". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1114–1122. ISBN: 9781450362016. DOI: 10.1145/3292500.3330896. URL: https://doi.org/10.1145/3292500.3330896.

[211] M. Veillette, S. Samsi, and C. Mattioli. "SEVIR : A Storm Event Imagery Dataset for Deep Learning Applications in Radar and Satellite Meteorology". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 22009–22019. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/fa78a16157fed00d7a80515818432169-Paper.pdf.

[212] R. Sluiter. *Interpolation methods for the climate atlas*. KNMI De Bilt, The Netherlands, 2012. URL: https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubTR/TR335.pdf.

[213] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. "Maskgit: Masked generative image transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11315–11325.

[214] R. Imhoff, C. Brauer, A. Overeem, A. Weerts, and R. Uijlenhoet. "Spatial and temporal evaluation of radar rainfall nowcasting techniques on 1,533 events". In: *Water Resources Research* 56.8 (2020), e2019WR026723.

[215] S. Coles. *An introduction to statistical modeling of extreme values*. Springer Series in Statistics. Springer-Verlag, 2001. ISBN: 1-85233-459-2.

[216] R. Gençay and F. Selçuk. "Extreme value theory and Value-at-Risk: Relative performance in emerging markets". en. In: *Int. J. Forecast.* 20.2 (Apr. 2004), pp. 287–303.

[217] L. De Haan and A. Ferreira. *Extreme Value Theory*. en. Springer Series in Operations Research and Financial Engineering. Springer Science+Business Media, Jan. 2006.

[218] S. Chen, N. Kalanat, S. Topp, J. Sadler, Y. Xie, Z. Jiang, and X. Jia. "Meta-transfer-learning for time series data with extreme events: An application to water temperature prediction". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. Birmingham United Kingdom: ACM, Oct. 2023.

[219] H. Tabari. "Climate change impact on flood and extreme precipitation increases with water availability". In: *Scientific reports* 10.1 (2020), p. 13768.

[220] Y. Zhang, M. Long, K. Chen, L. Xing, R. Jin, M. I. Jordan, and J. Wang. "Skilful nowcasting of extreme precipitation with NowcastNet". In: *Nature* (2023).

[221] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian. "Accurate medium-range global weather forecasting with 3D neural networks". In: *Nature* (2023).

[222] H. Bi, M. Kyryliuk, Z. Wang, C. Meo, Y. Wang, R. Imhoff, R. Uijlenhoet, and J. Dauwels. "Nowcasting of Extreme Precipitation Using Deep Generative Models". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[223] J. Jing, Q. Li, X. Ding, N. Sun, R. Tang, and Y. Cai. "AENN: A generative adversarial neural network for weather radar echo extrapolation". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2019), pp. 89–94.

[224] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar. "Integrating physics-based modeling with machine learning: A survey". In: *arXiv preprint arXiv:2003.04919* 1.1 (2020), pp. 1–34.

[225] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, *et al.* "Physics-informed machine learning: case studies for weather and climate modelling". In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200093.

[226] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, *et al.* "The ERA5 global reanalysis". In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049.

[227] B. S. Murphy. "PyKrige: development of a kriging toolkit for Python". In: *AGU fall meeting abstracts*. Vol. 2014. 2014, H51K–0753.

[228] M. A. Oliver and R. Webster. "Kriging: a method of interpolation for geographical information systems". In: *International Journal of Geographical Information System* 4.3 (1990), pp. 313–332.

[229] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[230] A. Daw, M. Maruf, and A. Karpatne. "PID-GAN: A GAN Framework based on a Physics-informed Discriminator for Uncertainty Quantification with Physics". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 237–247.

[231] Z. Pu and E. Kalnay. "Numerical weather prediction basics: Models, numerical methods, and data assimilation". In: *Handbook of hydrometeorological ensemble forecasting* (2019), pp. 67–97.

[232] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[233] R. V. Rohli and C. Li. *Meteorology for Coastal Scientists*. Springer Nature, 2021.

[234] K. P. Georgakakos and R. L. Bras. "A hydrologically useful station precipitation model: 1. Formulation". In: *Water Resources Research* 20.11 (1984), pp. 1585–1596.

[235] H. De Bruin. "From Penman to Makkink". In: *Evaporation and Weather: Technical Meeting 44, Ede, The Netherlands 25 March 1987. The Hague, Netherlands. 1987. p 5-31. 1 fig, 4 tab, 34 ref.* 1987.

[236] W. Yan, D. Hafner, S. James, and P. Abbeel. "Temporally Consistent Transformers for Video Generation". In: *arXiv preprint arXiv:2210.02396* (2022).

[237] C. Meo, A. Roy, M. Lică, J. Yin, Z. B. Che, Y. Wang, R. Imhoff, R. Uijlenhoet, and J. Dauwels. *Extreme Precipitation Nowcasting using Transformer-based Generative Models*. 2024. arXiv: 2403.03929 [cs.LG].

[238] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.* "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782 (2019), pp. 350–354.

[239] OpenAI. *OpenAI Five*. https://blog.openai.com/openai-five/. 2018.

[240] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.* "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (2016), pp. 484–489.

[241] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.* "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". In: *Science* 362.6419 (2018), pp. 1140–1144.

[242] M. Schmid, M. Moravcik, N. Burch, R. Kadlec, J. Davidson, K. Waugh, N. Bard, F. Timbers, M. Lanctot, Z. Holland, *et al.* "Player of games". In: *arXiv preprint arXiv:2112.03178* (2021).

[243] V. Micheli, E. Alonso, and F. Fleuret. "Transformers are sample-efficient world models". In: *arXiv preprint arXiv:2209.00588* (2022).

[244] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[245] D. Ha and J. Schmidhuber. "Recurrent world models facilitate policy evolution". In: *Advances in neural information processing systems* 31 (2018).

[246] J. Robine, M. Höftmann, T. Uelwer, and S. Harmeling. "Transformer-based world models are happy with 100k interactions". In: *International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=TdBaDGCpjly.

[247] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. *Language Models are Unsupervised Multitask Learners*. 2019.

[248] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21 (2020), pp. 1–67.

[249] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009.

[250] D. Liu, V. Shah, O. Boussif, C. Meo, A. Goyal, T. Shu, M. C. Mozer, N. Heess, and Y. Bengio. "Stateful Active Facilitator: Coordination and Environmental Heterogeneity in Cooperative Multi-Agent Reinforcement Learning". In: *The Eleventh International Conference on Learning Representations*. 2023.

[251] M. Janner, Q. Li, and S. Levine. "Offline Reinforcement Learning as One Big Sequence Modeling Problem". In: *Advances in neural information processing systems* 34 (2021).

[252] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. "Decision Transformer: Reinforcement Learning via Sequence Modeling". In: *Advances in neural information processing systems* 34 (2021).

[253] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, *et al.* "Model-based reinforcement learning for atari". In: *arXiv preprint arXiv:1903.00374* (2019).

[254] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. "Survey of hallucination in natural language generation". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38.

[255] D. Lee, C. Kim, S. Kim, M. Cho, and W. S. HAN. "Draft-and-revise: Effective image generation with contextual rq-transformer". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 30127–30138.

[256] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, J. Schrittwieser, K. Anderson, S. York, M. Cant, A. Cain, A. Bolton, S. Gaffney, H. King, D. Hassabis, S. Legg, and S. Petersen. *DeepMind Lab*. 2016. arXiv: 1612.03801 [cs.AI].

[257] R. Goyal, S. E. Kahou, V. Michalski, J. Materzyńska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. *The "something something" video database for learning and evaluating visual common sense*. 2017. arXiv: 1706.04261 [cs.CV].

[258] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. "Learning latent dynamics for planning from pixels". In: *arXiv preprint arXiv:1811.04551* (2018).

[259] W. L. Taylor. ""Cloze Procedure": A New Tool for Measuring Readability". In: *Journalism & Mass Communication Quarterly* 30 (1953), pp. 415–433. URL: https://api.semanticscholar.org/CorpusID:206666846.

[260] Z. Zhang, J. Ma, C. Zhou, R. Men, Z. Li, M. Ding, J. Tang, J. Zhou, and H. Yang. "M6-UFC: Unifying multi-modal controls for conditional image synthesis via non-autoregressive generative transformers". In: *arXiv preprint arXiv:2105.14211* (2021).

[261] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, A. G. Hauptmann, *et al.* "Language Model Beats Diffusion–Tokenizer is Key to Visual Generation". In: *arXiv preprint arXiv:2310.05737* (2023).

[262] H. Inan, K. Khosravi, and R. Socher. "Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=r1aPbsFle.

[263] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. *DeepMind Control Suite*. 2018. arXiv: 1801.00690 [cs.AI]. URL: https://arxiv.org/abs/1801.00690.

[264] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. 2018. arXiv: 1801.01290 [cs.LG]. URL: https://arxiv.org/abs/1801.01290.

[265] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. "Mastering Visual Continuous Control: Improved Data-Augmented Reinforcement Learning". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=_SJ-_yyes8.

[266] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[267] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. "Deep reinforcement learning at the edge of the statistical precipice". In: *Advances in neural information processing systems* 34 (2021), pp. 29304–29320.

[268] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. "Perplexity—a measure of the difficulty of speech recognition tasks". In: *The Journal of the Acoustical Society of America* 62.S1 (Aug. 2005), S63–S63. ISSN: 0001-4966. DOI: 10.1121/1.2016299. eprint: https://pubs.aip.org/asa/jasa/article-pdf/62/S1/S63/11558910/s63_5_online.pdf. URL: https://doi.org/10.1121/1.2016299.

[269] C. Meo, L. Mahon, A. Goyal, and J. Dauwels. "αTC-VAE: On the relationship between Disentanglement and Diversity". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=ptXo0epLQo.

[270] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. "Leveraging procedural generation to benchmark reinforcement learning". In: *International conference on machine learning*. PMLR. 2020, pp. 2048–2056.

[271] I. Kanitscheider, J. Huizinga, D. Farhi, W. H. Guss, B. Houghton, R. Sampedro, P. Zhokhov, B. Baker, A. Ecoffet, J. Tang, *et al.* "Multi-task curriculum learning in a complex, visual, hard-exploration domain: Minecraft". In: *arXiv preprint arXiv:2106.14876* (2021).

[272] R. Sullivan, A. Kumar, S. Huang, J. Dickerson, and J. Suarez. "Reward Scale Robustness for Proximal Policy Optimization via DreamerV3 Tricks". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 1352–1362. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/04f61ec02d1b3a025a59d978269ce437-Paper-Conference.pdf.

[273] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[274] R. Williams and J. Peng. "Function Optimization using Connectionist Reinforcement Learning Algorithms". English. In: *Connection Science* 3.3 (Jan. 1991), pp. 241–268. ISSN: 0954-0091. DOI: 10.1080/09540099108946587.

[275] R. J. Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8 (1992), pp. 229–256.

[276] G. N. DeSouza and A. C. Kak. "Vision for mobile robot navigation: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 24.2 (2002), pp. 237–267.

[277] A. Mahapatra and K. Kulkarni. "Controllable Animation of Fluid Elements in Still Images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 3667–3676.

[278] R. Ming, Z. Huang, Z. Ju, J. Hu, L. Peng, and S. Zhou. "A Survey on Video Prediction: From Deterministic to Generative Approaches". In: *arXiv preprint arXiv:2401.14718* (2024).

[279] D. Kondratyuk, L. Yu, X. Gu, J. Lezama, J. Huang, R. Hornung, H. Adam, H. Akbari, Y. Alon, V. Birodkar, *et al.* "Videopoet: A large language model for zero-shot video generation". In: *arXiv preprint arXiv:2312.14125* (2023).

[280] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long. "iVideoGPT: Interactive VideoGPTs are Scalable World Models". In: *Advances in Neural Information Processing Systems*. 2024.

[281] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 448–456.

[282] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. "Deconvolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2010, pp. 2528–2535.

[283] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[284] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. "Transformer-xl: Attentive language models beyond a fixed-length context". In: *arXiv preprint arXiv:1901.02860* (2019).

[285] D. Hafner. "Benchmarking the Spectrum of Agent Capabilities". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=1W0z96MFEoH.

[286] D. Yarats, I. Kostrikov, and R. Fergus. "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=GY6-6sTvGaf.

[287] M. Tan. "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents". In: *Readings in Agents*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 487–494. ISBN: 1558604952.

[288] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson. *Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?* 2020. eprint: arXiv:2011.09533.

[289] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson. *The StarCraft Multi-Agent Challenge*. 2019. arXiv: 1902.04043 [cs.LG].

[290] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *Neural Information Processing Systems (NIPS)* (2017).

[291] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games". In: *arXiv preprint arXiv:2103.01955* (2021).

[292] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, and Y. Yang. "Trust region policy optimisation in multi-agent reinforcement learning". In: *arXiv preprint arXiv:2109.11251* (2021).

[293] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, *et al.* "Starcraft ii: A new challenge for reinforcement learning". In: *arXiv preprint arXiv:1708.04782* (2017).

[294] J. Z. Leibo, E. A. Duéñez-Guzmán, A. S. Vezhnevets, J. P. Agapiou, P. Sunehag, R. Koster, J. Matyas, C. Beattie, I. Mordatch, and T. Graepel. "Scalable Evaluation of Multi-Agent Reinforcement Learning with Melting Pot". In: *ArXiv* abs/2107.06857 (2021).

[295] J. K. Gupta, M. Egorov, and M. Kochenderfer. "Cooperative multi-agent control using deep reinforcement learning". In: *International Conference on Autonomous Agents and Multiagent Systems*. Springer. 2017, pp. 66–83.

[296] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang. "Federated Reinforcement Learning with Environment Heterogeneity". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 18–37.

[297] K. Ndousse, D. Eck, S. Levine, and N. Jaques. "Emergent Social Learning via Multi-agent Reinforcement Learning". In: *ICML*. 2021.

[298] A. Goyal, A. Lamb, P. Gampa, P. Beaudoin, S. Levine, C. Blundell, Y. Bengio, and M. C. Mozer. "Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems". In: *International Conference on Learning Representations*. 2021.

[299] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf. "Recurrent Independent Mechanisms". In: *ArXiv* abs/1909.10893 (2021).

[300] N. Rahaman, M. W. Gondal, S. Joshi, P. V. Gehler, Y. Bengio, F. Locatello, and B. Scholkopf. "Dynamic Inference with Neural Interpreters". In: *NeurIPS*. 2021.

[301] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. "Continuous control with deep reinforcement learning". In: *CoRR* abs/1509.02971 (2016).

[302] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[303] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. "Counterfactual Multi-Agent Policy Gradients". In: *AAAI*. 2018.

[304] C. Li, C. Wu, T. Wang, J. Yang, Q. Zhao, and C. Zhang. "Celebrating Diversity in Shared Multi-Agent Reinforcement Learning". In: *arXiv preprint arXiv:2106.02195* (2021).

[305] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. F. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. "Value-Decomposition Networks For Cooperative Multi-Agent Learning". In: *ArXiv* abs/1706.05296 (2018).

[306] T. Rashid, M. Samvelyan, C. S. D. Witt, G. Farquhar, J. N. Foerster, and S. Whiteson. "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning". In: *ArXiv* abs/1803.11485 (2018).

[307] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang. "Qplex: Duplex dueling multi-agent q-learning". In: *arXiv preprint arXiv:2008.01062* (2020).

[308] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson. "MAVEN: Multi-Agent Variational Exploration". In: *NeurIPS*. 2019.

[309] C. Guestrin, M. Lagoudakis, and R. Parr. "Coordinated reinforcement learning". In: *ICML*. Vol. 2. Citeseer. 2002, pp. 227–234.

[310] L. Busoniu, R. Babuska, and B. De Schutter. "A comprehensive survey of multiagent reinforcement learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), pp. 156–172.

[311] S. Kapetanakis and D. Kudenko. "Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems". In: *Adaptive Agents and Multi-Agent Systems II*. Springer, 2004, pp. 119–131.

[312] Y.-C. Choi and H.-S. Ahn. "A survey on multi-agent reinforcement learning: Coordination problems". In: *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*. IEEE. 2010, pp. 81–86.

[313] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson. "Learning to Communicate with Deep Multi-Agent Reinforcement Learning". In: *NIPS*. 2016.

[314] S. Sukhbaatar, A. D. Szlam, and R. Fergus. "Learning Multiagent Communication with Backpropagation". In: *NIPS*. 2016.

[315] J. Jiang and Z. Lu. "Learning Attentional Communication for Multi-Agent Cooperation". In: *NeurIPS*. 2018.

[316] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau. "Tarmac: Targeted multi-agent communication". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1538–1546.

[317] W. Kim, W. Jung, M. Cho, and Y. Sung. "A maximum mutual information framework for multi-agent reinforcement learning". In: *arXiv preprint arXiv:2006.02732* (2020).

[318] Z. Xie and S. Song. "FedKL: Tackling Data Heterogeneity in Federated Reinforcement Learning by Penalizing KL Divergence". In: *arXiv preprint arXiv:2204.08125* (2022).

[319] F. Doshi-Velez and G. Konidaris. "Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations". In: *IJCAI: proceedings of the conference*. Vol. 2016. NIH Public Access. 2016, p. 1432.

[320] F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-28929-8. URL: https://doi.org/10.1007/978-3-319-28929-8.

[321] C. Zhang and V. Lesser. "Coordinating Multi-Agent Reinforcement Learning with Limited Communication". In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*. AAMAS '13. St. Paul, MN, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1101–1108. ISBN: 9781450319935.

[322] W. U. Mondal, M. Agarwal, V. Aggarwal, and S. V. Ukkusuri. "On the approximation of cooperative heterogeneous multi-agent reinforcement learning (marl) using mean field control (mfc)". In: *Journal of Machine Learning Research* 23.129 (2022), pp. 1–46.

[323] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. J. Henaff, M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira. "Perceiver IO: A General Architecture for Structured Inputs & Outputs". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=fILj7WpI-g.

[324] F. Christianos, G. Papoudakis, A. Rahman, and S. V. Albrecht. "Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing". In: *ICML*. 2021.

[325] J. K. Terry, N. Grammel, A. Hari, L. Santos, and B. Black. "Revisiting parameter sharing in multi-agent deep reinforcement learning". In: *arXiv preprint arXiv:2005.13625* (2020).

[326] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The graph neural network model". In: *IEEE Transactions on Neural Networks* 20.1 (2008), pp. 61–80.

[327] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[328] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. "Neural relational inference for interacting systems". In: *arXiv preprint arXiv:1802.04687* (2018).

[329] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.* "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[330] D. Liu, A. Lamb, K. Kawaguchi, A. Goyal, C. Sun, M. C. Mozer, and Y. Bengio. "Discrete-Valued Neural Communication". In: *ArXiv* abs/2107.02367 (2021).

[331] M. Shanahan. "A cognitive architecture that combines internal simulation with a global workspace". In: *Consciousness and cognition* 15.2 (2006), pp. 433–449.

[332] S. Dehaene, H. Lau, and S. Kouider. "What is consciousness, and could machines have it?" In: *Science* 358.6362 (2017), pp. 486–492.

[333] A. Goyal, A. Didolkar, A. Lamb, K. Badola, N. R. Ke, N. Rahaman, J. Binas, C. Blundell, M. Mozer, and Y. Bengio. "Coordination among neural modules through a shared global workspace". In: *arXiv preprint arXiv:2103.01197* (2021).

[334] W. Kim, J. Park, and Y. Sung. "Communication in Multi-Agent Reinforcement Learning: Intention Sharing". In: *International Conference on Learning Representations*. 2020.

[335] Y. Wang, F. Zhong, J. Xu, and Y. Wang. *ToM2C: Target-oriented Multi-agent Communication and Cooperation with Theory of Mind*. 2021. arXiv: 2111.09189 [cs.MA].

[336] M. A. Weis, K. Chitta, Y. Sharma, W. Brendel, M. Bethge, A. Geiger, and A. S. Ecker. *Unmasking the Inductive Biases of Unsupervised Object Representations for Video Sequences*. 2020. arXiv: 2006.07034 [cs.CV].

[337] V. I. Zhukovskiy and K. N. Kudryavtsev. "Pareto-optimal Nash equilibrium: Sufficient conditions and existence in mixed strategies". In: *Automation and Remote Control* 77.8 (2016), pp. 1500–1510.

[338] M. Bowling and M. Veloso. *An analysis of stochastic game theory for multiagent reinforcement learning*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2000.

[339] K. Boutoustous, G. J. Laurent, E. Dedu, L. Matignon, J. Bourgeois, and N. Le Fort-Piat. "Distributed control architecture for smart surfaces". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 2018–2024.

[340] L. Matignon, G. J. Laurent, and N. Le Fort-Piat. "Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems". In: *The Knowledge Engineering Review* 27.1 (2012), pp. 1–31.

[341] L. Zheng, J. Yang, H. Cai, W. Zhang, J. Wang, and Y. Yu. "MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence". In: *CoRR* abs/1712.00600 (2017). eprint: 1712.00600. URL: http://arxiv.org/abs/1712.00600.

[342] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.

[343] C. Team. *Chameleon: Mixed-Modal Early-Fusion Foundation Models*. 2024. arXiv: 2405.09818 [cs.CL]. URL: https://arxiv.org/abs/2405.09818.

[344] C. Glymour, K. Zhang, and P. Spirtes. "Review of causal discovery methods based on graphical models". In: *Frontiers in Genetics* 7 (2016), p. 4.

[345] J. Pearl and D. Mackenzie. *The book of why: The new science of cause and effect*. Basic Books, 2018.

[346] J. Woodward. *Making things happen: A theory of causal explanation*. Oxford University Press, 2003.

[347] J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

[348] H. Lachapelle, A. Goyal, Y. Bengio, and G. Lajoie. "Gradient-based neural dag learning". In: *arXiv preprint arXiv:2306.04013* (2023).

[349] K. Chalupka, P. Perona, and F. Eberhardt. "Visual causal reasoning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 700–708.

[350] G. Brockman. *GPT-4 Technical Report*. https://openai.com/research/gpt-4. 2023.

[351] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.* "Evaluating large language models trained on code". In: *arXiv preprint arXiv:2107.03374* (2021).

[352] A. Chowdhery, S. Narayanan, J. Kwiecinski, J. Bastings, M. Littman, K. Srinivasan, R. Agarwal, S. Sabour, A. Tiwary, G. Tussing, *et al. Palm 2 technical report*. Tech. rep. Google, 2023.

CRISTIAN MEO

cmeo97.github.io @ C.Meo@tudelft.nl +31 06 11 800165 cristian-meo Cmeo97

RESEARCH

PhD in Generative AI

Technische Universiteit Delft

October 2021 – Present Delft, NL

- I am PhD student in CS at TUDelft advised by **Prof. Justin Dauwels** and **Prof. Geert Leus**. I'm working on Generative Modeling, Unsupervised Representation Learning and Model based RL.

Research Intern at MILA

Mila - Quebec AI Institute

February 2023 – July 2023 Montreal, CA

- Supervised by **Prof. Yoshua Bengio** (A.M. Turing Award 2018) and **Anirudh Goyal** (RS@Google DeepMind). I worked on three projects:
  - Hierarchical World Models (in collaboration with Alex Zakharov)
  - On the relationship between Disentanglement and Diversity.
  - Object-Centric World Models (in collaboration with Aniket Didolkar)

EDUCATION

Master Degree in Bio Robotics

Technische Universiteit Delft

September 2019 – August 2021 Delft, NL

- **Graduated cum Laude - Honours Program**: top 2% students rank.
- **Research Intern at Donders, Radboud University, NL**, where I worked on **Multimodal-VAE Active Inference Controller**, published at IROS2021 https://arxiv.org/pdf/2103.04412.

Bachelor degree in Mechanical Engineering

Polytechnic of Turin

September 2016 – July 2019 Turin, IT

- **Graduated with 110/110**.
- **Participated in the Young Talent program**: top 2% students rank.
- **SEC-U Exchange Scholarship Program at Auburn University**

SELECTED PUBLICATIONS

- **Masked Generative Priors Improve World Models Sequence Modelling Capabilities**, Best Paper at ICLR2025 World Models Workshop https://arxiv.org/pdf/2410.07836.
- **Object-Centric Temporal Consistency via Conditional Autoregressive Inductive Biases**, Poster at NeurIPS2024 Workshop: Compositional Learning, https://arxiv.org/pdf/2410.15728.
- **Extreme Precipitation Nowcasting using Transformer-based Generative Models**, Spotlight at ICLR 2024 Workshop: Tackling Climate Change with Machine Learning, https://arxiv.org/html/2403.03929v1.
- **On the relationship between Disentanglement and Diversity**, Poster at ICLR2024, https://openreview.net/pdf?id=ptXo0epLQo, (project partially done at Mila).

SKILLS

Deep Learning (PyTorch, Tensorflow, JAX), Generative AI, NLP, Multimodal Learning, LLMs, Unsupervised and Object-centric Representation Learning, GNNs, PEFT, Model-based RL, World Models, Robot dynamics and control, ROS, C/C++, Python, Matlab, UNIX. Team Lead/Management, Public Speaking.

AWARDS

Surf Computational Grant

Snellius: the Dutch National Supercomputer
400.000€ of computational resources.

Infineon's IPCEI PhD Booster

Mobility Sponsorship of **20.000€**

Research Intern, MILA Scholarship

Feb 2023 - July 2023 Montreal, CA

Talent Program

Polytechnic of Turin
Top 2% students.

TUDelft Honours Program

Top 2% students.

OTHER EXPERIENCES

Mentoring/Team Lead

Technische Universiteit Delft, NL

October 2022 Delft, NL

Since the beginning of my PhD I supervised over **25 Master students** and mentored **4 PhD students**.

Deep Learning Instructor

Technische Universiteit Delft, NL

Oct 2022 - April 2025

Deep Learning of EE4685 Machine learning, a Bayesian perspective.

REFEREES

Prof. Justin Dauwels (PhD Supervisor)
@ J.H.G.Dauwels@tudelft.nl
HB 17.070, TUDelft

LANGUAGES

Italian	**Native**
English	**Advanced (C1/C2)**
French	**Elementary (A2)**

PUBLICATIONS

Below is a list of all papers I contributed to that have been published along my PhD.

1. **Stateful active facilitator: Coordination and environmental heterogeneity in cooperative multi-agent reinforcement learning**
   The Eleventh International Conference on Learning Representations (ICLR 2023), 2022

2. **TC-VAE: On the relationship between Disentanglement and Diversity**
   The Twelfth International Conference on Learning Representations (ICLR 2024), 2023

3. **Nowcasting of extreme precipitation using deep generative models**
   ICASSP 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, 2023

4. **Bayesian-LoRA: LoRA based Parameter Efficient Fine-Tuning using Optimal Quantization levels and Rank Values through Differentiable Bayesian Gates**
   ICML24 Workshop on Advancing Neural Network Training (WANT), 2024

5. **Extreme Precipitation Nowcasting using Transformer-based Generative Models**
   Spotlight acceptance, ICLR 2024 Workshop: Tackling Climate Change with Machine Learning, 2024

6. **Object-centric temporal consistency via conditional autoregressive inductive biases**
   NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and, 2024

7. **Precipitation Nowcasting Using Physics Informed Discriminator Generative Models**
   European Conference on Signal Processing (EUSIPCO 2024), 2024

8. **Image Search Engine by Deep Neural Networks**
   Pre-Proceedings of the 2022 Symposium on Information Theory and Signal Processing, 2022

9. **Masked Generative Priors Improve World Models Sequence Modelling Capabilities**
   Best Paper Award,ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling, 2025

10. **Discrete Messages Improve Communication Efficiency among Isolated Intelligent Agents**
    arXiv preprint arXiv:2312.15985, 2023