Sharpening the Future of Occupancy Grid Map Prediction Methods

R.F. Dirks

MSc Thesis

Delft University of Technology

Sharpening the Future of Occupancy Grid Map Prediction Methods

An Investigation into Loss Functions and Semantic Segmentation Multi-Task learning for More Accurate OGM Predictions

by

R.F. Dirks

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday June 13th, 2022 at 3PM.

Student Number:	4464877	
Institution:	Delft University of Technology	
Date:	Sunday 5th June, 2022	
Thesis Committee:	H.J. Boekema	TU Delft, daily supervisor
	Prof. Dr. D.M. Gavrila	TU Delft, supervisor
	Dr. J.F.P. Kooij	TU Delft
Previous daily supervisor:	Dr. E.A.I. Pool	

Cover Image: Motion Prediction Visualization by Rutger Dirks

TUDelft

Acknowledgement

The research that comprised the content of this Master thesis is performed at the Delft University of Technology, the faculty of 3mE, the department of Cognitive Robotics, at the Intelligent Vehicles section. The objective of this thesis is to propose two methods that improve the quality of motion prediction methods using Occupancy Grid Maps within the domain of intelligent vehicles.

During the period I was working on this thesis, I learned to overcome many challenges both academically and personally, where the effects of the COVID-19 pandemic prompted me to be resourceful and resilient to continue the research.

This thesis would not have been possible without the support, supervision and constructive feedback from Hidde J-H. Boekema, Dr. Ir. Ewoud A.I. Pool, and Prof. Dr. Dariu M. Gavrila, for which I would like to express my gratitude. Besides my supervisors, I would like to thank my family, my former roommates and friends for their support and advice during my Master. I also would like to thank the people from the Intelligent Vehicles section for their advice, discussions, and kindness.

Lastly, I am grateful for my time as board member of SAVE/Asimov and Co-creator of the Msc. Robotics during which I hoped to contribute to a greater academic experience for future students. I am looking forward to the next chapter in life.

R.F. Dirks
Delft, June 2022

Abstract

For an Autonomous Vehicle (AV) to traverse safely in traffic, It is vital it can anticipate the behavior of surrounding traffic participants using motion prediction. Current motion prediction approaches can be categorized into object-centered and object-agnostic methods and are primarily based on deep learning. The former relies on a human-engineered pipeline of object detection and tracking, of which the errors can accumulate in the motion predictions. The latter does not rely on this pipeline; however, it lacks the ability to learn object representations causing blurriness and object disappearances for longer-term predictions which forms a safety hazard.

This thesis proposes two methods to improve the performance of the object-agnostic sequence-to-sequence Occupancy Grid Map (OGM) prediction networks, trained on the Waymo Open Perception dataset. The first method uses inter-pixel loss functions, i.e. the SSIM and Sinkhorn losses, instead of the ubiquitously used per-pixel losses, to train the PredRNN++ Occupancy Grid Map (OGM) prediction network. Inter-pixel losses take into account the spatial relations between grid cells during the evaluation of OGMs, whereas per-pixel losses evaluate each grid cell's value independently. The quantitative results demonstrate that using inter-pixel losses can improve short term predictions with a prediction horizon of $T = 5$ for the Mean Squared Error (MSE) by 4.3%, Image Similarity (IS) by 7.8%, Average Precision (AP) by 0.3%, metrics. For the longer term, $T = 15$, the predictions improve for the MSE by 5.0%, IS by 20.5%, AP by 0.1%, and Accuracy by 0.6%. Furthermore, the use of inter-pixel losses reduces blurriness and object disappearances. The second method is based on multi-task learning. By training the PredRNN++ to perform the prediction task together with the semantic segmentation task on the predicted OGMs, it is expected to learn object representations which it uses to improve the prediction quality. The quantitative results show that multi-task learning does not improve the OGM predictions. However, some qualitative results show that multi-task learning reduces blurriness and object disappearances.

Contents

Nomenclature

AAConvLSTM Attention Augmented Convolutional Long Short-Term Memory (ConvLSTM). vii, 19–21, 53

AP Average Precision. ii, xi, xii, 27, 28, 38, 40–42, 44, 45, 48, 49, 51, 60, 62, 63, 65–67

AV Autonomous Vehicle. ii, vi, vii, 1–5, 10, 13, 14, 22, 47, 51–53

BBA Basic Belief Assignment. xi, 6, 7

BEV bird's-eye view. 3, 5, 13–15, 53

cdf cumulative distribution function. viii, 32–35, 48

CDNA Convolutional Dynamic Neural Advection. 21

CNN Convolutional Neural Network. vii, 13–15, 17, 21, 52

ConvLSTM Convolutional Long Short-Term Memory. v, vii, 14, 15, 17–19, 21, 37, 40, 53

DOGMa Dynamic Occupancy Grid Map. vii, 14, 15, 17, 20, 23

DST Dempster-Shafer Theory. vi, 5–7

ECP2.5D Eurocity Persons 2.5D. 10, 11, 52

FCN Fully Convolutional Network. 19

FOD Frame of Discernment. 6

GHU Gradient Highway Unit. vii, 24, 25, 37, 40

GRU Gated Recurrent Unit. vii, 14, 17

IS Image Similarity. ii, xi, xii, 27, 28, 38, 40–42, 44, 45, 47–49, 51, 53, 60, 62, 63, 65–67

KITTI Karlsruhe Institute of Technology and Toyota Technological Institute. 10, 11, 18–20, 50

LSTM Long Short-Term Memory. vii, 17, 20, 21, 24, 25

MFE Motion-Flow Extraction. vii, 18

MSE Mean Squared Error. ii, viii, xi, xii, 21, 22, 27, 33–35, 38, 40, 42, 44, 45, 47, 48, 51, 53, 60, 62, 63, 65–67

OGM Occupancy Grid Map. ii, iii, vi–ix, xi, 3–12, 14–30, 32, 35–38, 40, 41, 44–53, 60, 63–66

pdf probability density function. viii, 32–35, 48

PR Precision-Recall. 28

RNN Recurrent Neural Network. vii, 14, 17, 18

SSIM Structural Similarity Index Measure. iii, vi, viii–xi, 4, 5, 8, 9, 22, 24–26, 29–32, 35–53, 59–62, 65, 66, 68, 69

STC Spatio-Temporal Convolution. vii, 15, 16

STM Spatial Transformer. vii, 17

VRU Vulnerable Road User. 1, 2, 13

List of Figures

List of Tables

1 Introduction

Based on research by the World Health Organization, more than half of the fatalities caused by road traffic crashes are among Vulnerable Road Users (VRUs) such as pedestrians and (motor-)cyclists [65]. Therefore, the United Nations General Assembly has set the target of halving the global number of deaths and injuries from road traffic crashes from approximately 1.3 million to 0.65 million by 2030. A US study on vehicle crashes by The National Motor Vehicle Crash Causation Survey found that the critical reason for 94% of the traffic accidents could be attributed to the driver [1]. Consequently, a primary reason for the development of AVs is to erase the effects of human error and thereby increase road safety [11]. In the past few years, the development of self-driving vehicles has been booming, as is demonstrated by the recent deployment of the world's first fully autonomous taxi service by Waymo [28] and with Cruise's robotaxi service announced to commence in the coming months [2]. These ride-hailing services are the beginning of a new age of autonomous transport. However, there are still safety and security concerns that need to be addressed to deploy AVs on a global scale. Unfortunately, in 2018 the first fatal crash has occurred between an AV and a pedestrian, after which it was argued that AVs cannot yet accurately predict human behavior [36]. Cui et al. [11]'s review, including safety failures of AVs, states that AVs need to interact and cooperate with other road users, including vehicles, cyclists, and pedestrians, to ensure safety in the future. Interaction and cooperation between AVs and road users require the ability to detect the road users and anticipate their behavior so a safe and timely reaction can be carried out.

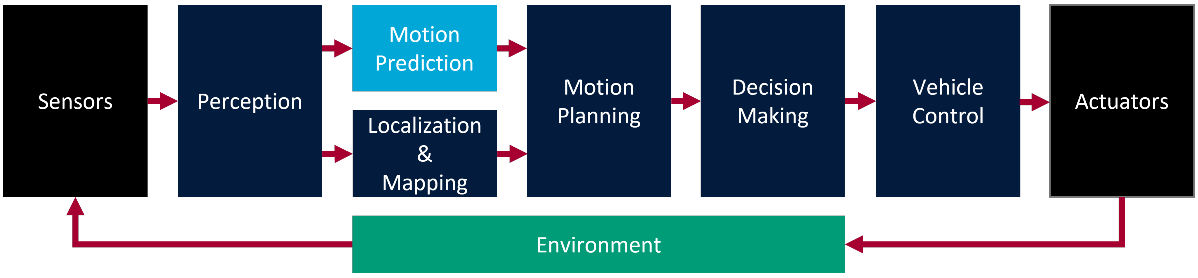
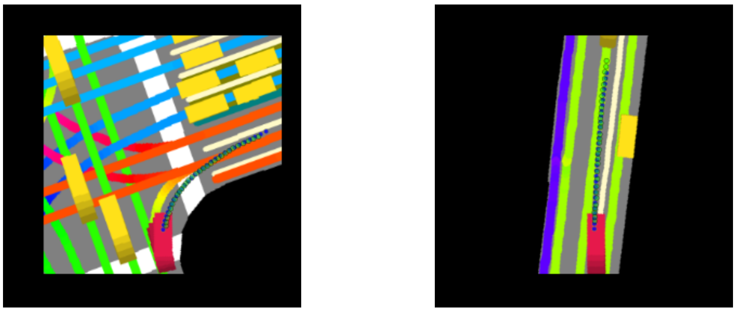


Figure 1.1: A diagram example of the Autonomous Navigation Process. The sensors and actuators (black) measure and act on the environment (green), respectively. The cognitive processing steps (dark and light blue) interpret the sensor data and make vehicle control decisions based on those interpretations. This thesis focuses on the Motion Prediction cognitive processing step (light blue).

One paradigm of the autonomous vehicle navigation process is a loop that starts by measuring the environment with sensors, followed by a cognitive process in which vehicle control decisions are made. Those decisions are executed on the environment by the vehicle's actuators. Figure 1.1 shows a diagram example of this navigation process. Between the sensors and actuators, a cognitive process takes place in which the measurements are interpreted (perception), which allows the environment to be mapped and anticipated (localization & mapping, motion prediction) so the possible paths can be planned to manoeuvre through the environment (motion planning). The decision making step picks the path that best fits the vehicle's goal criteria (e.g. going towards the destination, following traffic rules, keeping safe distances), after which the vehicle control step computes how the actuators must be controlled to follow the intended path. This navigation process is repeated continuously while the AV is driving. In this thesis, the focus lies on improving the motion prediction step.

Currently, most motion prediction methods use a form of machine learning. The earlier methods found in literature were based on classical Bayesian machine learning methods. More recently, the application of deep learning models increased due to developments in computation power and deep learning libraries, which made them more feasible and accessible. The deep learning based methods outperform the more classical machine learning approaches because of the large number of trainable parameters and increased generalizability the former can accomplish. The motion prediction methods can be subdivided into two categories which will be called object-centered and object-agnostic prediction in this thesis.

In object-centered prediction, object features obtained from a pipeline consisting of object detection and tracking are implemented in the prediction method. The future states for each detected object are predicted, given their past state information and information about other detected features from the environment such as other road users, lane markings, signs, and static objects [32, 68, 34, 53, 16, 9, 43, 26, 42, 37]. Figure 1.2 shows an example of object-centered motion prediction.



(a) Crossing **(b)** Straight road

Figure 1.2: Example predictions of Djuric et al. [16]'s object-centered motion prediction model. The model inputs are the object's (red) past velocity, acceleration, and rotation rate, and the past raster images containing road (rgb-colors dependent on direction) and object (yellow) information (figures a and b). The model outputs are the object's positional state predictions for 3s into the future. Those predictions are projected onto the raster image using blue dots, with the ground truth using green dots. a) shows an example near a crossing. b) shows an example on a straight road.

An advantage of object-centered prediction is that for each detected and tracked object a prediction can be made. Furthermore, by using object detection methods, not only objects' locations can be measured, but also motion prediction features such as object class, object velocity, and object orientation can be obtained. This can improve the prediction accuracy [34, 32, 68]. Figure 1.3 visually demonstrates that road user detection is performed accurately on the Waymo Open Perception Dataset [60], which is large-scale real-world dataset for AV research. Subsequently, predicting the future locations of those road users is an essential part of the safe deployment of AVs. However, this is a challenging task because road users, specifically VRUs, have complex and mutable motion patterns which depend on the objects and other road users in their environment as stated by [9].

As a result, a disadvantage of object-centered methods is that they are dependent on the accuracy of the object detection and tracking pipeline. For instance, Luo, Yang, and Urtasun [42]'s state-of-the-art object-centered prediction method uses an object detection and tracking pipeline that has an object detection and tracking recall of 92.5%, which means that 7.5% of the objects are not considered for motion prediction. Moreover, for state-of-the-art tracking methods, it is still a challenge to track VRUs in urban environments. In dense areas, tracks get lost or switch identities, and velocity estimates become less accurate [21]. Another disadvantage is that object-centered prediction methods only predict objects' future states and not the entire environment's occupancy states. If the AV fails to consider the future states of the environment, including the objects in it, its behavior can have profound safety implications.

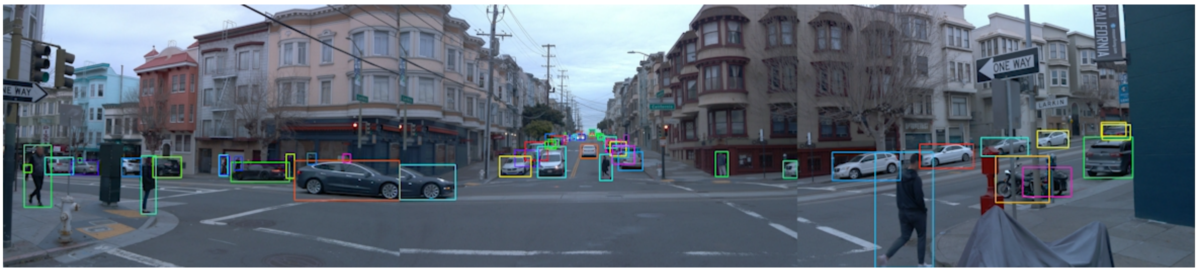
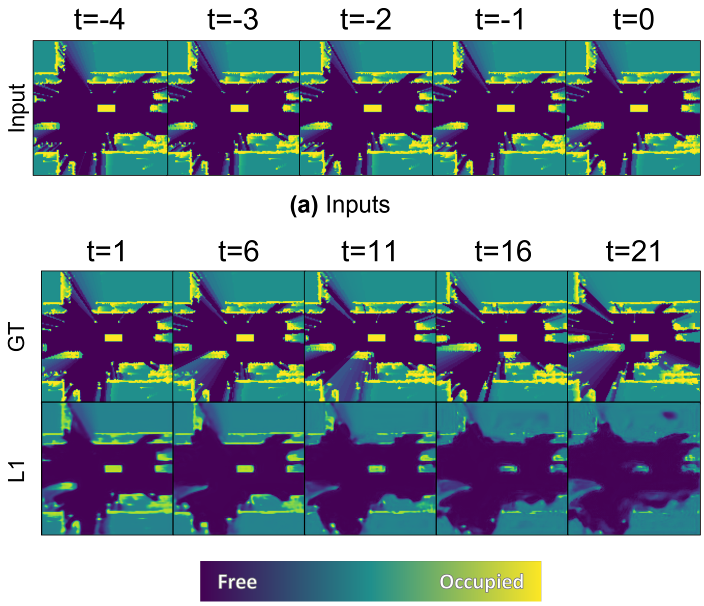


Figure 1.3: The Waymo Autonomous Vehicle (AV)'s view with labeled road users as recorded in one of the large-scale real-world datasets for AV research, i.e. the Waymo Open Perception dataset [60].

In object-agnostic prediction, on the other hand, no human-engineered pipeline is required for motion prediction. Environment representations are generated that lie close to raw sensor data compared to the object-centered methods. These environment representations are predicted for future time steps using deep learning [27, 35, 61, 25, 57, 15, 44, 66]. A popular representation in literature is the Occupancy Grid Map (OGM), which is a 2D bird's-eye view (BEV) occupancy map of the AV's environment. An OGM is a tensor or matrix in which each element's value represents the occupancy state of a corresponding spatial area in the AV's environment. They are directly generated from the raw sensor data without performing processing steps such as object detection and tracking. An in-depth explanation on OGMs is provided in section 1.4.1, where figure 1.5 shows an example of an OGM. Using OGMs is advantageous because it makes object-agnostic prediction independent from the accuracy and representations of object-centered object detection and tracking steps. Although height information is lost because 3D objects are reduced to a 2D bird's-eye view (BEV) representation, the image-like representation makes OGMs suitable to easily extract spatial features from using convolutions in a deep learning network. Given a sequence of past OGMs, the network then outputs future OGMs. This way, the future occupancy states for the entire area around the AV are predicted, unlike object-centered prediction, where the predictions are limited to a selection of detected objects. An example of object-agnostic OGM prediction is shown in figure 1.4. The disadvantage, however, of object-agnostic prediction is that without object detection and tracking, it is difficult to incorporate motion prediction features in the environment representations to benefit the prediction accuracy. Consequently, it is challenging for object-agnostic prediction methods to distinguish objects from the rest of the environment. A result is that occupied regions related to objects will disappear or merge with the environment (blur out) as these regions are not recognized as separate, rigid bodies as the environment's uncertainty increases in longer-term predictions [27, 35, 61]. One example of the predictions becoming increasingly blurry and objects disappearing as the time horizon increases can be seen in figure 1.4.

Among the prediction methods found in literature, a trade-off is made between the long-term accuracy of the predictions (object-centered prediction) and to what extent the AV's environment is considered in the prediction (object-agnostic prediction). In other words, one can state it is a quality-quantity trade-off. Various state-of-the-art prediction methods aim to overcome this trade-off by incorporating environment occupancy information in the object-centered approaches or by incorporating object information in object-agnostic approaches. For instance, [16, 9, 26, 42, 37] each incorporate the environment's occupancy states in their prediction models to improve their object-centered predictions. Considering object-agnostic prediction methods, the proposed solutions found in literature mitigate the blurriness and object disappearances for longer-term predictions by implementing mechanisms for the network to learn class and object representations of the environment. For example, [35] propose an attention-based network architecture that enables their network to more easily access temporal and spatial information to learn and highlight different class representations in the OGM, which reduces blurriness and object disappearances. Furthermore, [25, 57, 66, 61] incorporate dynamic information about the objects in the OGMs, while [44, 66] incorporate semantic class information in the OGMs to have their networks learn the behavior of objects in the OGMs.

Despite the developments in both object-centered and object-agnostic prediction methods, overcoming the quality-quantity trade-off is still a challenge. This thesis focuses on overcoming this trade-off



(a) Inputs

(b) Outputs

Figure 1.4: An example of object-agnostic OGM predictions generated using PredRNN++ [62] on a test example from the Waymo Perception dataset [60] for the network trained with the L1 loss function. a) An input of 5 past frames is provided as input to the network. b) predictions are made for 20 future frames (every fifth frame shown). Ground truth future frames are shown in the top row of b).

in the category of object-agnostic prediction using OGMs by aiming to improve the accuracy of longer-term predictions. The following section describes the problem from which the research in this thesis was initiated.

1.1. Problem Formulation

The problem of object-agnostic motion prediction using OGMs is a self-supervised sequence-to-sequence inference task. Based on evidence in the form of past OGMs generated from the AV's environment measurements, future OGMs are predicted for a specified time horizon. Within this broader problem lies the issue that for longer prediction horizons, the predicted OGMs become blurrier, and objects disappear. The resulting problem is that the OGM prediction must learn object representations to reduce the blurriness and object disappearance.

1.2. Research Questions

This thesis investigates whether there are methods that can be applied during the training process of the OGM prediction network to increase its ability to learn object representations to reduce the blurriness and object disappearance in its predictions.

Current literature shows that most prediction methods use the ubiquitous per-pixel losses to train the OGM prediction network [27, 35, 61, 25, 57, 15]. These per-pixel losses evaluate similarity between a prediction and the ground truth by individually comparing each grid cell corresponding to the same respective location. Therefore per-pixel losses ignore spatial relations between grid cells when evaluating similarities. Opposite to the per-pixel loss, the term 'inter-pixel loss' is introduced in this thesis, under which all losses fall that evaluate similarities between a prediction and the ground truth by considering the relations between multiple grid cells within the OGMs and comparing those relations. This term is introduced because the investigated literature does not mention a term that encompasses losses such as the SSIM [64] and the Sinkhorn losses [13] which can evaluate high-level properties such as the structure and style of an image by considering (spatial) relations between grid cells. The first research question therefore states:

1. *Does the use of an inter-pixel loss function, such as the Structural Similarity Index Measure loss or the Sinkhorn loss, to train a sequence-to-sequence Occupancy Grid Map prediction network improve its performance, reduce blurriness and objects disappearance compared to using a per-pixel loss?*

This thesis hypothesizes that using an inter-pixel loss will train the network to learn and maintain spatial distance relations between grid cells. As a result, it is expected that the network will recognize clustered occupied grid cells as objects. It will learn the difference between occupied regions and free space and, in turn, will sustain them in the predictions, reducing blurriness and object disappearance.

Further, research by [15] demonstrates that their object tracking and OGM prediction network learns latent representations of spatial and dynamic patterns of different object classes from the tracking task, which the network can utilize to improve the semantic segmentation task of OGMs. This principle of inductive transfer learning, specifically multi-task learning, could be applied to improve the quality of OGM predictions as well. This leads to the second research question:

2. *Does the performance of an Occupancy Grid Map prediction network improve by means of multi-task learning if it also learns to perform semantic segmentation on the predictions?*

The expectation is that a network can learn a latent object recognition to improve the occupancy prediction task, i.e. predicting a sequence of future OGMs, if it is trained to also perform semantic segmentation on those OGMs. This thesis hypothesizes that by training an OGM prediction network to also predict semantic labels for each grid cell in the future OGMs, it will learn latent object representations related to the semantic classes of those grid cells. Subsequently, the network can use the latent object representations to improve the prediction task by reducing the disappearance of objects and blurriness.

1.3. Contributions

The key contributions of this thesis are as follows:

- To improve the performance of a sequence-to-sequence OGM prediction network by using the SSIM and Sinkhorn-L1 composite inter-pixel, distance capturing losses,
- To investigate the effects of multi-task learning the occupancy prediction task and semantic segmentation task on the performance of a sequence-to-sequence OGM prediction network.

1.4. Preliminaries

This thesis investigates two ways to improve Occupancy Grid Map (OGM) prediction. One method considers the use of inter-pixel loss functions. The other method is based on multi-task learning of the occupancy prediction task and the semantic segmentation task on OGMs. This section provides background information about what this thesis defines as OGMs, followed by information about loss functions, then the explanation of semantic OGM prediction, and closing with the explanation of multi-task learning.

1.4.1. Occupancy Grid Map

In this thesis, object-agnostic prediction considers the prediction of OGMs. OGMs are 2D bird's-eye view (BEV) environment maps of the AV's environment, generated directly from the raw sensor data it has obtained (e.g. from LiDAR, radar, or camera measurements). Elfes [17] introduced this environment mapping method for robust mobile robot perception and navigation purposes. An OGM is a grid that consists of cells that each represent the occupancy of a spatial area in the AV's environment with a specified resolution. Probabilistic sensor models are used to maintain estimates of the occupancy state of each cell, which allows uncertainties to be mapped in the grid. Probabilistic OGMs represent each cell's occupancy with a continuous value between 0 and 1. 0 means that the area is empty, and 1 means that the area is occupied, where a cell's complete uncertainty of the occupancy state is assigned by the value 0.5. For binary OGMs, each grid cell's value is rounded to either 0 or 1. In this thesis, the probabilistic OGMs are generated using the Dempster-Shafer Theory (DST) as sensor model described by [47] followed by the concept of pignistic probability as performed by [35].

This thesis refers to the latter OGMs as evidential OGMs. An example of such an OGM can be seen in figure 1.5. The following subsections explain how the DST and the principle of pignistic probability are used to generate the OGMs

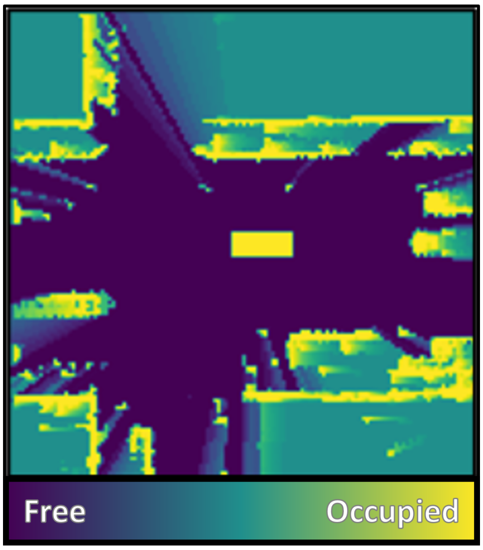


Figure 1.5: An OGM of size 128x128 with a resolution of 33x33 centimeter per grid cell generated from LiDAR measurements from the Waymo Open Perception dataset [60], using the Dempster-Shafer Theory (DST) [47] and pignistic probability as performed by [35]. Free, occupied, and uncertain grid cells are indicated by the dark blue, yellow, and green color hues respectively.

The Dempster-Shafer Theory

The Dempster-Shafer Theory (DST) formalizes the transferable belief model, which is a model that defines a discrete Frame of Discernment (FOD) which contains the set of possible states of a system. In the case of an OGM, the FOD is $\Omega = \{E, O\}$, with E for Empty and O for Occupied. A mass function M is defined that maps the powerset 2^{Ω} of Ω to the domain $[0\ 1]$. The powerset is the set of all subsets of Ω including the empty set $\emptyset$ and itself ($2^{\Omega} = \{\emptyset, E, O, \Omega\}$). If A is an element in 2^{Ω}, then $M(A)$ represents the amount of evidence (mass) that supports hypothesis A within the domain $[0\ 1]$. Then, two properties are set to the mass function M. First, $M(\emptyset) = 0$, and second, formula 1.1 verifies that the sum of the masses of each hypothesis in the powerset is equal to 1. This means that it is assumed that the powerset 2^{Ω} is complete and that no evidence will be obtained that supports none of the hypotheses in the FOD. A mass function with these two properties is called a Basic Belief Assignment (BBA) mass function. [45]. A BBA mass function allows for the assignment of lower and upper bounds of a probability interval, belief $Bel()$ and plausibility $Pl()$ respectively, that represent the support to the hypotheses $A \in 2^{\Omega}$.

Belief is the sum of the masses of all the hypothesis's subsets, including the hypothesis, as is shown in equation 1.2, where A and B represent hypotheses (e.g. for the Ω-hypothesis, this would be the mass of Ω and the subsets of Ω: E and O]). In equation 1.3 it can be seen that the plausibility is computed by taking 1 minus the sum of the masses that exclude the hypothesis. For instance, for the hypothesis O, the plausibility $Pl(O)$ would be 1 minus $M(\emptyset)$ and $M(E)$ which is 0.3. [45].

$$\sum_{A \in 2^{\Omega}} M(A) = 1 \tag{1.1}$$

$$Bel(A) = \sum_{B | B \subset A} M(B) \tag{1.2}$$

$$Pl(A) = \sum_{B | B \cap A \neq \emptyset} M(B) = 1 - \sum_{B | B \cap A = \emptyset} M(B) \tag{1.3}$$

An example of a BBA mass function is shown in table 1.1, given the powerset $2^{\Omega} = \{\emptyset, E, O, \Omega\}$. In this example, a sensor reading obtains information that a certain grid cell is empty. The sensor reading's probability of being reliable is 0.7 and 0.3 for being unreliable. This information can be used to assign a subjective probability (mass) to each hypothesis, which sums up to 1. A reliable sensor will give a true reading, so the hypothesis of the grid cell being empty ($m(E)$) is assigned a mass of 0.7. However, given that there *is* a grid cell ($m(\emptyset) = 0$), the sensor reading has a probability of 0.3 that it is unreliable. This does not mean that the grid cell is occupied with a probability of 0.3, but it means that its state is uncertain with that probability. Therefore, a mass of 0.3 is assigned to the hypothesis $m(\Omega)$ that states that the grid cell is either empty or occupied. The hypothesis of the cell being occupied ($m(O)$) will have a mass of 0, since no evidence supports it [59]. Subsequently, the belief $Bel()$ and the plausibility $Pl()$ of the hypotheses are computed, giving the lower and upper probability bounds for the hypotheses.

Table 1.1: An example of a BBA mass function.

Hypothesis	Mass	Belief	Plausibility
$M(\emptyset)$	0	0	0
$M(E)$	0.7	0.7	1.0
$M(O)$	0	0	0.3
$M(\Omega)$	0.3	1.0	1.0

Applying DST to generate OGMs
To generate an evidential OGM, given independent sensor data, each grid cell is assigned a mass function $M_{i,j}$ with beliefs and plausibilities. If new sensor data about a grid cell is obtained, the DST method can fuse the mass functions of the current cell state and the new sensor data according to the joint mass equations 1.4 and 1.5 [45].

$$M_1 \oplus M_2(A) = \left\{ \begin{array}{ll} \frac{M_{1 \cap 2}(A)}{1 - M_{1 \cap 2}(\emptyset)} & A \neq \emptyset \\ 0 & A = \emptyset \end{array} \right\} \tag{1.4}$$

Where $\cap$ is the conjunctive rule with B and C hypotheses and A the joint hypothesis:

$$M_{1 \cap 2}(A) = \sum_{B,C \in 2^{\Omega} | A = B \cap C} M_1(B) \cdot M_2(C) \tag{1.5}$$

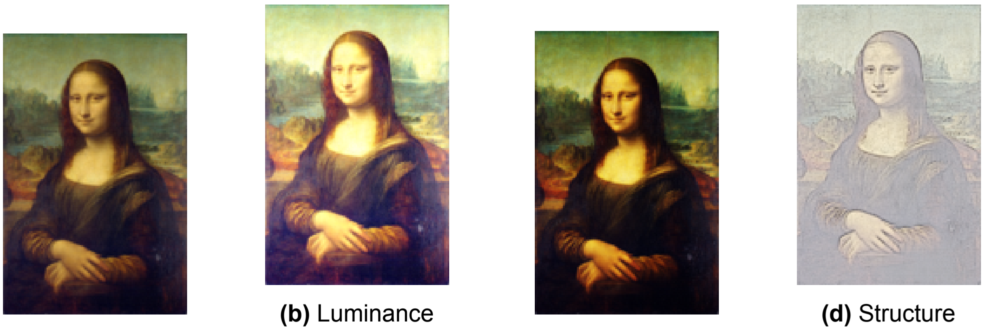
Then, a probability measure can be taken from the mass function using the principle of pignistic probability according to equation 1.6. Here, A and B represent hypothesis subsets of powerset 2^{Ω}. The cardinality (the number of elements in a subset) is denoted as two vertical bars.

$$P(A) = \sum_{B \in 2^{\Omega}} M(B) \cdot \frac{|A \cap B|}{|B|} \tag{1.6}$$

It should be noted that this equation is not bijective, meaning an infinite number of mass functions can be found given the same probability. This is because the information that distinguishes ignorance from uncertainty is lost when the mass function is transformed into a probability.

1.4.2. Loss functions

In the domain of deep learning, the loss function determines the error between a prediction and the ground truth. Subsequently, the gradient is taken from the loss function to perform gradient descent to optimize the deep learning network's weights (backpropagation). For loss functions to be suitable for deep learning practices, they must be differentiable so a gradient can be derived to backpropagate the error through the deep learning network. The loss function that is used influences which properties of the prediction are rewarded and which are penalized and thus how a network's weights are updated. Therefore, the loss should evaluate the similarity between the ground truth and its prediction to achieve the desired network optimizations. However, there are multiple ways to define similarity. In the case of OGMs, similarity can mean that each individual grid cell in the prediction must be as close in value as the ground truth's cells. Measuring this similarity can be achieved by per-pixel loss functions such as the L1 loss. Still, using a per-pixel loss has resulted in increasingly blurry predictions and object disappearances for longer time horizons. Alternatively, similarity can be measured by evaluating how multiple grid cells relate to each other within a predicted OGM and comparing those relations to the relations between grid cells in the ground truth OGM. This can be achieved by inter-pixel loss functions, which is a term introduced in section 1.2 in this thesis. An example of an inter-pixel loss function is the SSIM loss [63, 64], of which the workings are intuitively shown in figure 1.6. By using an inter-pixel loss, it is expected that the network better learns to recognize (spatially) related grid cells such as object instances and the difference between occupied and free space compared to using a per-pixel loss. If these relations are learned better, it is expected that the network generates more certain predictions which contain less blur and fewer object disappearances. This thesis investigates whether the use of inter-pixel loss functions, i.e. the SSIM and Sinkhorn [13] losses, to train an OGM prediction network result in an improved prediction performance compared to using a per-pixel loss such as the L1 loss.



(a) Ground truth

(b) Luminance (intensity)

(c) Contrast (variance)

(d) Structure (correlation)

Figure 1.6: A visually intuitive illustration on Leonardo Da Vinci's Mona Lisa of the workings of the inter-pixel SSIM loss [63, 64]. The SSIM is based on how humans recognize similarities between images. Humans mainly perceive differences based on b) luminance, c) constrast and d) structure, compared to the ground truth a). Mathematically, an image's luminance, contrast, and structure difference can be defined as the mean pixel intensity value, the pixel value's variance, and the correlation between the prediction and ground truth respectively. Figure adjusted and taken from C2RMF Retouched [6].

1.4.3. Semantic Occupancy Grid Map prediction

OGMs only contain information about the occupancy states of the environment. This occupancy information can be extended with semantic information. Semantic OGM prediction is the prediction of semantic labels for each grid cell in the predicted OGMs. This can be done by training a network to predict OGMs with additional semantic channels that each represent a semantic class. For every grid cell in each semantic channel, a value between 0 and 1 is assigned that represents the probability of the corresponding grid cell in the OGM belonging to the channel's semantic class. An example of semantic OGM prediction by [41] is shown in figure 1.7.

1.4.4. Multi-task learning

Multi-task learning is a sub-category of transfer learning, where transfer learning is a method used in machine learning in which previously learned knowledge to perform one task is retained and reused to perform new tasks better [50]. For instance, there are two domains, a target domain D_T and a source domain D_S of training data, e.g. occupancy data for OGM predictions and semantic label data for semantic segmentation of the OGMs respectively, used to perform the target task T_T and source task T_S, e.g. the OGM prediction task and the semantic segmentation task of OGMs respectively. Transfer learning is a method that aims to improve learning the target task $T - T$ from the domain D_T using the knowledge gained from learning the source task T_S from domain D_S. In the case of multi-task learning, both the target and source task are learned simultaneously, given the training data of both the target and source domains. When training a model to learn both tasks simultaneously, the expectation is that

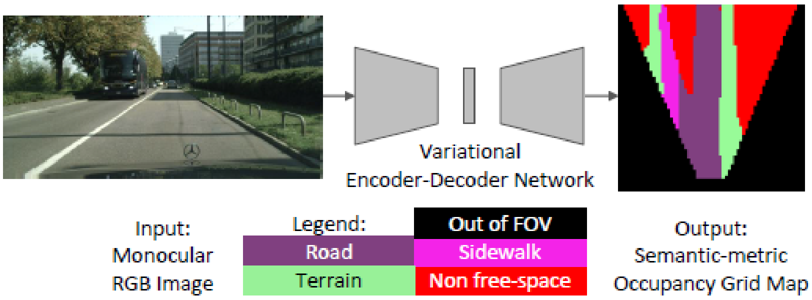


Figure 1.7: An illustration of the semantic OGM method that takes a monocular RGB image as input and outputs a semantic OGM, taken from [41]. Lu, Molengraft, and Dubbelman [41] distinguish the environment with four labels: road, sidewalk, terrain, and non-free space.

learning the source task has a regularizing effect on learning the target task by adjusting the model's parameters to be useful for both tasks, preventing the parameters to overfit on the target task. The effect is that this improves the model's performance on the target task [50].

1.5. Submitted Article

In addition to this thesis, a paper is submitted that complements this thesis's research. The paper: *"An Experimental Study on Occupancy Grid Map Prediction Loss Functions"* by Hidde J-H. Boekema, Rutger F. Dirks, and Dariu M. Gavrila presents an experimental study on various loss functions used in OGM prediction. The effects of the ubiquitous L1 and L2 loss functions, and composites thereof, together with the Binary Cross-Entropy loss and inter-pixel SSIM loss are investigated in a OGM setting using the PredRNN++ [62] network. The L1 loss function generally achieves the best performance on the metrics, where some qualitative evidence is found that the SSIM and composite L1L2 and SmoothL1 losses result in reduced blurriness and fewer disappearing dynamic objects. The full paper is provided in Appendix D.

1.6. Thesis Outline

The outline of this thesis is as follows. Chapter 2 covers the related work in which relevant literature in the domain of datasets, motion prediction, video prediction, loss functions, semantic segmentation, and multi-task learning are discussed. Chapter 3 describes the method that is used to answer the two research questions posed in section 1.2. Following the method, Chapter 4 reports and discusses the OGM prediction experiments and obtained results from this research. The experiments are followed by a thorough discussion of the conducted research in Chapter 5, the conclusion of this thesis in Chapter 6, and future work possibilities in Chapter 7.

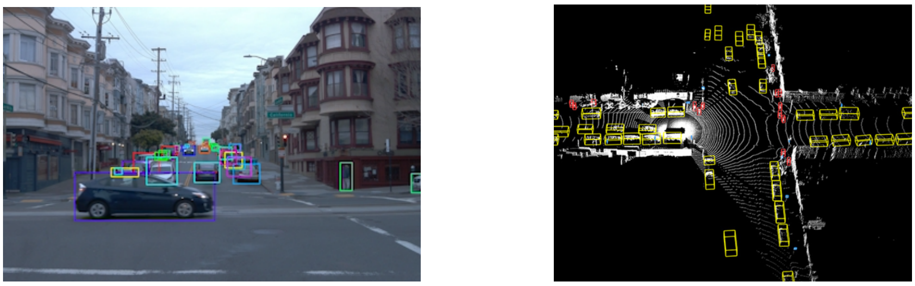
2

Related Work

The foundation of this thesis research is built upon the related work described in this section. First, the datasets suitable for object-agnostic OGM prediction are described. This is followed by highlighting relevant object-centered and object-agnostic prediction methods. Subsequently, several video prediction methods are highlighted that form the basis for many object-agnostic methods. Next, a section describes the relevant literature about the use of different loss functions in the domain of OGM prediction, followed by a section about semantic segmentation of OGMs. The final section describes earlier research on multi-task learning for OGM prediction networks.

2.1. Datasets

To generate OGMs for motion prediction in traffic, a dataset is required that contains traffic scene sequences consisting of dense occupancy measurements around the AV. Any occupancy measurements, whether it is from a LiDAR, radar, stereo cameras, ultrasonic sensors, or a combination of those [17], can be used to generate OGMs. OGMs are a versatile means to represent the environment because they can be generated using one or a combination of sensors. Usually, LiDAR measurements are used as raw data for OGM generation because LiDARs have a large, high-resolution measurement range in the order of 100 meters around the AV. Besides occupancy data, the dataset should also contain class labels linked to the occupied regions. The class labels are necessary for the evaluation of the semantic segmentation task. The datasets that meet these requirements are the Waymo Perception dataset [60], the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset [22], the nuScenes dataset [7], the Eurocity Persons 2.5D (ECP2.5D) dataset [5], and the Argoverse dataset [8]. This section describes these respective datasets together with table 2.1 which gives an overview of each dataset.



(a) A front camera image with object annotations from the Waymo [60] dataset.

(b) The annotated objects in the LiDAR point cloud from the Waymo [60] dataset.

Figure 2.1: Two examples of data annotation in the Waymo [60] dataset.

The Waymo Open Perception dataset was originally recorded in 3 cities (Phoenix, San Francisco, and Mountain View). It contains 1150 20-second sequences at 10 Hz with 10.8M objects with tracking IDs, labels for four object classes (vehicles, pedestrians, cyclists, and signs), and 3D bounding boxes of each of those objects. Moreover, [35] and [61] used the Waymo dataset to generate and predict OGMs, so there is research to compare prediction methods with on this dataset. The KITTI dataset is the smallest dataset compared to the other datasets that are suitable for OGM prediction. It only contains 22 sequences and is obtained in only one country. The scenes in this dataset have a large diversity of urban and suburban scenes.This dataset has been used by [27], [44], and [35] to generate and predict OGMs. Therefore, there is research to compare prediction methods with that use the KITTI dataset. For the nuScenes dataset, a LiDAR at 20 Hz, five radars at 13 Hz, and six 1600x900 resolution cameras around the vehicle at 12 Hz are used to collect data. However, the sampling frequency of the data is only 2 Hz. This is the lowest sampling frequency compared to other datasets. Figure 2.2 shows an example of the six camera images overlayed with the LiDAR point cloud of the nuScenes dataset. The advantage of this dataset is that it contains velocity information from the radar data. Besides that, the nuScenes dataset is collected for object detection as well as object tracking. It contains a 1000 20 second sequences. Objects of 23 classes are annotated, including vehicles, pedestrians, and riders. The nuScenes dataset is larger and more diverse than most other datasets.

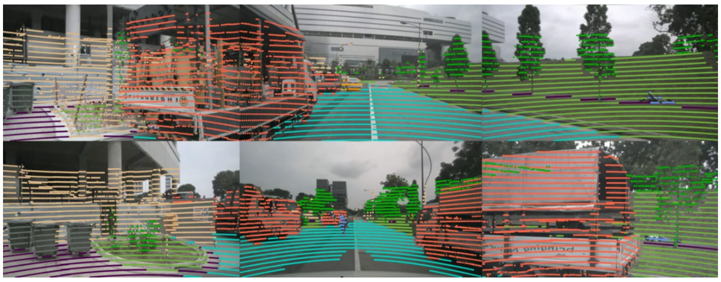


Figure 2.2: An example of the images of the six cameras from the nuScenes dataset [7]. The images are overlayed with the semantic segmented LiDAR data.

The ECP2.5D dataset [5] contains 15-second sequences in which seven different classes are labeled. This dataset is recorded in 31 cities in 12 countries, making it the most diverse dataset compared to the other ones regarding geographic locations. Also, the sampling frequency of the sequences is 20 Hz, which is the highest among the datasets. The Argoverse dataset [8] is recorded in only two cities in one country. It contains little geographic diversity compared to the other datasets. On the other side, the Argoverse dataset contains 333K 5-second sequences and labels 17 different classes, including vehicles, pedestrians and cyclists.

Dataset	Year	Sensors			Sampling Frequency	Location Diversity	Dataset size	Label Diversity	# Pedestrians	# Vehicles	OGM generation	OGM prediction
Waymo [60]	2020	5 LiDAR sensors (1 mid range, 4 short range)	5 cameras (3 front of res 1920x1280 and 2 side of res 1920x1014)	-	10 Hz	1 country, 3 cities, urban and suburban areas	1150 20s sequences	4 classes	~2.8M	~6.1M	[35] [61]	[35] [61]
KITTI [22]	2012	Velodyne HDL-64 LiDAR with 100m range (10 Hz)	4 cameras (2 color 2 grayscale) of res 1392x512 (10 Hz)	GPS/IMU	10 Hz	1 country, 1 city, dense urban areas	22 sequences	2 categories, 8 classes	~9.4K	~100K	[27] [44] [35]	[27] [44] [35]
NuScenes [7]	2019	32 beam LiDAR 1.4M points/s (20 Hz) + 5 Radars with 250m range (13 Hz)	6 cameras of res 1600x900 cameras (12 Hz)	GPS/IMU, CAN bus data	2 Hz	2 countries, 2 cities, dense urban areas	1000 20s sequences	23 classes, 8 attributes	~200K	~500K	[55] [40]	-
ECP2.5D [5]	2020	Velodyne HDL-64E LiDAR with 120m range (5-20 Hz)	1 RGB camera of res 1920x1024 (20 Hz)	GPS/INS	20 Hz	12 countries, 31 cities, dense urban areas	15s sequences, 46K frames	7 classes	~218K	-	-	-
Argoverse [8]	2019	2 VLP-32 LiDARs with 200m range (10 Hz)	7 RGB cameras in 360 deg setup of res 1920x1200 (30 Hz) and 2 in stereo view of res 2056x2464 (5 Hz)	GPS/ Odometry sensor	10 Hz	1 country, 2 cities, dense urban areas	~333K 5s sequences, 1006h of recording, spanning ~290km	17 classes	~1.5K	~8K	[55]	-

Table 2.1: An overview of the datasets that are suitable for OGM prediction. The columns ’OGM generation’ and ’OGM prediction’ list relevant literature that previously generated OGMs and/or performed OGM prediction with the corresponding dataset, respectively.

2.2. Object-centered prediction

In object-centered traffic prediction, a pipeline is designed manually that uses the AV's raw sensor data to perform object detection and object tracking to obtain object trajectories. These trajectories are used to predict the future states of these objects. In turn, the predictions are used by a motion planner to guide the future trajectory of the AV [42]. Object-centered prediction can be subdivided into classical object-centered prediction and deep learning based object-centered prediction. The former uses classical machine learning approaches such as Kalman Filters and Dynamic Bayesian Networks to perform predictions. The latter uses deep learning methods. Both object-centered methods are discussed in this section.

2.2.1. Classical object-centered prediction

Earlier methods used classical machine learning approaches to predict road user behavior. For instance, [32] propose two optical flow based methods, the Gaussian process dynamical model, and probabilistic hierarchical trajectory matching, for pedestrian path prediction in road crossing scenarios. First, an object detection module is used to generate bounding boxes around pedestrians from which it extracts positional data about the pedestrian. Then, the positional information and the extracted optical flow features are used to predict pedestrian trajectories that extend to future time steps to anticipate whether the pedestrian will cross the street. Kooij et al. [34] build upon VRU prediction methods by extracting motion cues from the static and dynamic context of the object of interest as well as the object context itself, such as head orientation and hand gestures, to improve the prediction accuracy of a switching linear dynamical system. Figure 2.3 shows an example of [34]'s proposed motion cues. Also, [53] propose the use of a mixture of linear dynamic systems to predict future trajectories of cyclists given the cyclists' past trajectory. They use road topology as environmental information to enhance the predictions.

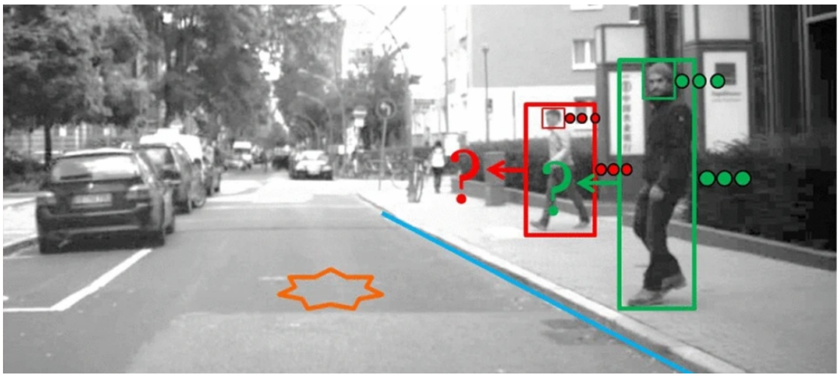


Figure 2.3: An example of object-centered motion prediction. The detected pedestrians (in the larger red and green bounding boxes) are analyzed to extract their gazing direction from their heads (in the smaller red and green bounding boxes). This information is used, together with the distance to the curb side (blue line) and other environment factors to predict whether the pedestrians will cross the street or not [34].

2.2.2. Deep learning based object-centered prediction

The downside of the classical machine learning models is that they are not generalizable for real-time applications. Deep learning methods have been demonstrated to be a successful alternative to efficiently perform motion prediction whilst obtaining environmental features encoded from the predicted objects' contexts [10]. An example is Cui et al. [10]'s multimodal motion prediction network that uses a deep CNN. For each detected object in the AV's environment, the past states of that object as well as a bird's-eye view (BEV) rasterized image of the object's context, which includes road and surrounding object information, are provided to the network. The network is trained to predict multiple future trajectories per object with the respective probabilities per trajectory. Furthermore, Djuric et al. [16] and Chou et al. [9] both perform object-centered prediction using rasterized high-definition maps that contain the surroundings of the glsAV including surrounding actors and lane markings and directions.

Implementing information about the environment, demonstrates that the object predictions become more accurate than without the environment information. The former, [16], presents a way to predict the uncertainties together with the predicted states. The latter, [9], focuses on pedestrian prediction in crowded urban scenes. Both methods use a CNN to perform the predictions. Moreover, Ma et al. [43] proposes a method that uses interaction modeling to analyze the interactions between different road users. This information is used to improve their predictions. Li et al. [37] builds upon this concept by proposing the Interaction Transformer architecture, which combines lane graphs, road masks, camera images, LiDAR maps, and occupancy data to extract features that represent the AV's environment and road user interactions.

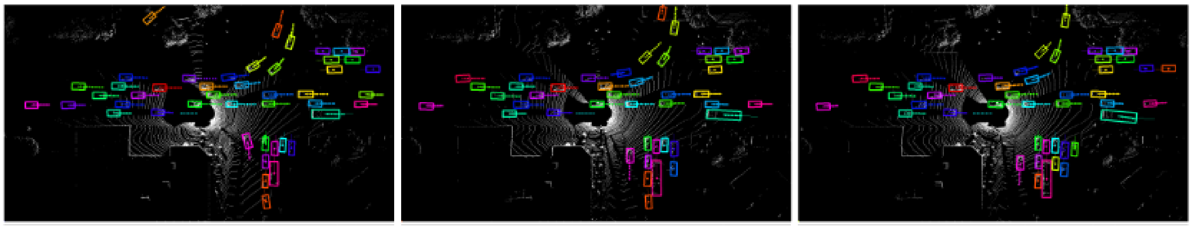


Figure 2.4: Results from [42]'s end-to-end motion prediction model. Each frame represents a different time step in which each vehicle has its own identifying color. The 'dots' represents the vehicle's current and predicted future center locations.

However, Luo, Yang, and Urtasun [42] states that choosing a cascade approach, where motion prediction is preceded by separate object detection and object tracking steps, can result in failures due to the accumulating errors in each step from which the downstream processes cannot recover. Instead of a cascade approach, Luo, Yang, and Urtasun [42] propose a deep learning based method that combines those three steps into one end-to-end motion prediction model using 3D sensor data directly. An example of [42]'s results is shown in figure 2.4. Given multiple past timesteps of a 3D voxel representation of an AV's environment, they train a CNN to perform object detection, followed by tracking of those objects and motion prediction. The downside of this method is that motion prediction can only be performed on the detected and tracked objects and evaluated on the true positives. If errors occur anywhere in the network's detection and tracking pipeline, they would accumulate and result in erroneous predictions.

The trend in object-centered prediction approaches is to enhance the predictions by implementing more of human-engineered environmental features, such as road lane information, interactions with other road users, and environment raster images. An alternative is to train a network to independently extract and use environmental features on an environment representation that only requires minimal pre-processing. This is done in object-agnostic prediction methods, which are described next.

2.3. Object-agnostic prediction

With the increasing possibilities of deep learning, the object-agnostic approach to motion prediction gained popularity. The object-agnostic representations do not require a human-engineered pipeline, consisting of object detection and tracking, but can be generated by minimally processing the raw sensor data. A popular used representation is an Occupancy Grid Map (OGM), of which a more in-depth description is provided in section 1.4.1. Less common representations are often extensions of the OGM with additional channels that contain dynamic (DOGMa) or semantic information [25, 57, 27, 61] for each grid cell such as velocities and orientations, and semantic class labels respectively. Alternatively, [66] creates multi-channel BEV rasters from LiDAR sweeps at different heights per channel. Each channel also contains semantic class and dynamic information per occupied raster cell. This section elaborates on several object-agnostic prediction methods. The methods are categorized based on the kind of neural network architecture used. First, two CNN based methods are described followed by an RNN based method using a GRU architecture. Then, two methods that use the ConvLSTM, a combination of a CNN and an RNN, are described. This section ends with three methods based on the PredNet [39] architecture, which implements ConvLSTM modules, followed by table 2.2 which shows an overview of the main properties of each object-agnostic prediction method.

2.3.1. CNN-based OGM predictors

Hoermann, Bach, and Dietmayer [25] proposes to predict Dynamic Occupancy Grid Map (DOGMa) using a Convolutional Neural Network (CNN). DOGMas are OGMs that contain dynamic information, i.e. velocities and orientations, for each grid cell. The network requires one DOGMa of the current time step and predicts up to 3 seconds into the future with time steps of 0.5 seconds. Only one input DOGMa is used because [25] hypothesizes that most information necessary for prediction can be found in the dynamic representation and the relation between the cells and not necessarily from the past DOGMas. By weighting dynamic grid cells relatively more than static grid cells in the loss, the network is prevented from ignoring the dynamic grid cells in its predictions. The output of the network is a multi-channel OGM with one channel containing the occupancy of static grid cells and several channels, for each predicted time step one, containing the future occupancy of the dynamic grid cells. The network's architecture is a downscaling cascade of CNNs followed by upscaling deconvolutions. It is based on Noh, Hong, and Han [46] semantic segmentation network (see figure 2.5) but it replaces the unpooling layers with deconvolutions. This is because deconvolution kernels can be learned, allowing more parameters to generate predictions. Hoermann, Bach, and Dietmayer [25]'s network is trained on their own recorded dataset. The proposed method performs multimodal predictions, can distinguish static from dynamic objects, and provides more accurate predictions than the particle filter baseline. This is because the convolutional layers capture the DOGMa's cell dependencies where particle filters assume independent cells. However, due to uncertainty, longer-term predictions become increasingly blurry.

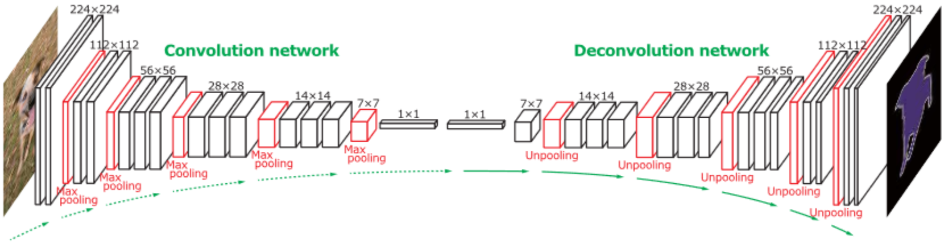


Figure 2.5: Noh, Hong, and Han [46] CNN architecture that forms the basis for Hoermann, Bach, and Dietmayer [25] DOGMa prediction network. The network's architecture is a downscaling cascade of CNNs followed by upscaling deconvolutions. In [25]'s architecture, the unpooling layers are replaced by deconvolution layers.

Another CNN based network is [66]'s MotionNet, which is a BEV raster prediction method. Wu, Chen, and Metaxas [66] performs perception and motion prediction with 3D LiDAR data as input. A sequence of LiDAR 3D point cloud sweeps synchronized to the current time frame is converted to BEV maps with semantic class and dynamic information by voxellizing the point cloud and representing it as a 2D pseudo-image where the height dimension corresponds to the image channels. Hence, it is similar to having a multi-channel pseudo-OGM where each channel represents a different height in the environment. This representation makes convolution possible. The MotionNet is a spatio-temporal pyramid network, as is shown in figure 2.6. It contains an encoder with multiple STC blocks that consist of standard 2D convolutions for capturing spatial features, followed by a 1D convolution. The encoder is followed by a decoder using deconvolutions. There are temporal pooling skip-connections between the encoder and decoder that convey temporal features at different scales to retain both global and local spatio-temporal contexts. The network contains three output heads, each providing an output with specific information. The first head provides semantic segmentation of the pseudo-OGM. The second head provides an pseudo-OGM prediction for a time horizon of N time steps. The third head classifies a grid cell as either dynamic or static. Together, the three heads from a multi-channel pseudo-OGM output with object labels, predictions, and dynamic information. This is similar to a sequence of DOGMas with semantic information.

The network is trained on the nuScenes [7] dataset. The MotionNet network [66] is compared with Schreiber, Hoermann, and Dietmayer [57]'s ConvLSTM encoder-decoder network, and four other baselines, which MotionNet all outperforms. Especially because of its ability to distinguish static and dynamic obstacles well. Figure 2.7 shows some of the predictions made using MotionNet.
All in all, the current OGM prediction methods still struggle with blurriness of longer-term predictions.

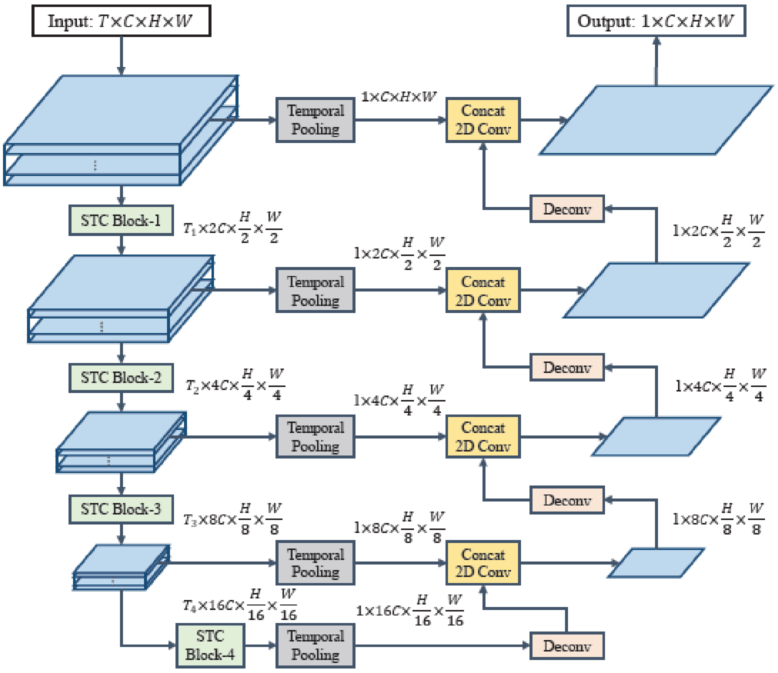

Input: $T \times C \times H \times W$
Output: $1 \times C \times H \times W$
Temporal Pooling
$1 \times C \times H \times W$
Concat 2D Conv
Deconv
STC Block-1
$T_1 \times 2C \times \frac{H}{2} \times \frac{W}{2}$
$1 \times 2C \times \frac{H}{2} \times \frac{W}{2}$
STC Block-2
$T_2 \times 4C \times \frac{H}{4} \times \frac{W}{4}$
$1 \times 4C \times \frac{H}{4} \times \frac{W}{4}$
STC Block-3
$T_3 \times 8C \times \frac{H}{8} \times \frac{W}{8}$
$1 \times 8C \times \frac{H}{8} \times \frac{W}{8}$
$T_4 \times 16C \times \frac{H}{16} \times \frac{W}{16}$
$1 \times 16C \times \frac{H}{16} \times \frac{W}{16}$
STC Block-4

Figure 2.6: An overview of MotionNet's architecture, [66]'s spatio-temporal pyramid network containing STC building blocks and temporal pooling skip connections to predict pseudo-OGMs.

The following section will illustrate that this challenge is not bound to OGM prediction only but also persists in the domain of video prediction methods, which often form the basis for OGM prediction.

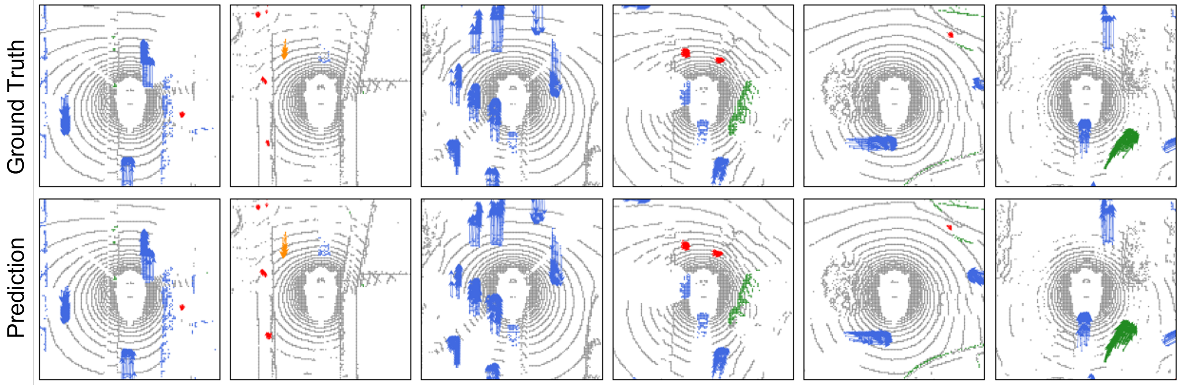


Figure 2.7: Wu, Chen, and Metaxas [66]'s pseudo-OGM predictions based on the LiDAR pointcloud sweeps containing dynamic and semantic class information. The image shows six separate predictions (bottom) with the corresponding ground truths (top). The background is gray, vehicles blue, pedestrians red, cyclists orange, and others are green.

2.3.2. RNN-based OGM predictor

Dequaire et al. [15]'s main research objective is, given a sequence of partially observable OGMs, to track the observed objects and predict the true, unoccluded state of the world in terms of current and future OGMs. Dequaire et al. [15] generates two-channel OGMs. One channel contains the occupancy data (0 for Free, 1 for Occupied), and the other contains visibility information (0 for occluded, 1 for unoccluded). Dequaire et al. [15] assumes that if a network learns to predict traffic behavior, it must learn to distinguish the shapes and motion patterns of the observed traffic participants. With this knowledge, the network can also be trained to perform semantic segmentation if class labels are provided in the OGMs during network training (multi-task learning). Different object classes (i.e. pedestrians, cyclists, vehicles) have distinct shapes and motion patterns. To perform the tasks of tracking, prediction, and semantic segmentation, [15] proposes a network based on Ondruska and Posner [48]'s recurrent deep tracking network, which contains RNNs. However, to improve this network's memory [15] uses the Gated Recurrent Unit (GRU) in their network architecture (see figure 2.8). A GRU is a network that retains memory for a longer term compared to an RNN. Furthermore, to compensate for the ego-motion in between an OGM sequence's time steps, [15] implements a Spatial Transformer (STM). The STM is a learnable module that uses odometry data to translate the GRU's hidden state of the previous time step to the current time step. The network is trained on [15]'s self-recorded dataset. The results show that [15]'s GRU-based network performs better [48]'s Deep Tracking network, which is used as a baseline.

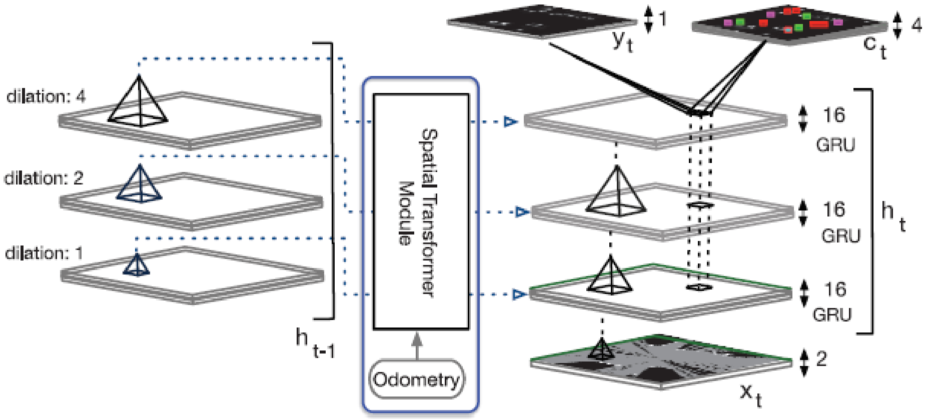


Figure 2.8: Dequaire et al. [15]'s GRU-based OGM prediction network architecture. The latent features from the previous time step go through the Spatial Transformer (STM) to compensate for the ego-motion in between the OGM frames.

2.3.3. ConvLSTM-based OGM predictors

This section elaborates on two OGM prediction methods based on the ConvLSTM [67] architecture, which consists of a CNN part and an LSTM part. The CNN part of the ConvLSTM is efficient in obtaining spatial features from image-like data structures. Subsequently, the LSTM [24] is a type of RNN architecture that can process sequential data and keeps track of long-term dependencies in the features obtained from the data sequences, such as OGM sequences. First, Schreiber, Hoermann, and Dietmayer [57]'s method is discussed which predicts DOGMas. Then, Mohajerin and Rohani [44]'s method is discussed, which proposes a so-called difference learning architecture for the prediction of OGMs.

Schreiber, Hoermann, and Dietmayer [57] builds upon [25]'s research by expanding the CNN with a ConvLSTM encoder and decoder containing ConvLSTMs and with ConvLSTM skip-connections between the down-scaling and up-scaling parts of the network (see figure 2.9). The ConvLSTMs in the encoder and decoder capture spatial and temporal correlations. The skip-connections convey high-resolution features, including occluded objects, to the up-scaling layers. Like [25]'s network, [57]'s network also weighs the static and dynamic predictions separately. The network outputs a multi-channel OGM. One channel contains the occupancy of static grid cells, and for every predicted time step, there are channels with the occupancy of the dynamic grid cells. To train the network, [57] recorded their own dataset. The network is compared with [25]'s and [15]'s networks and a particle filter. Schreiber, Hoermann, and Dietmayer [57]'s method shows a better performance. However, the data is only recorded in a stationary scenario. The performance on dynamic scenarios is not evaluated.

Mohajerin and Rohani [44] builds upon [15]'s research and proposes a difference learning method for the prediction of OGMs using ConvLSTMs in the network architecture. Figure 2.10 shows the

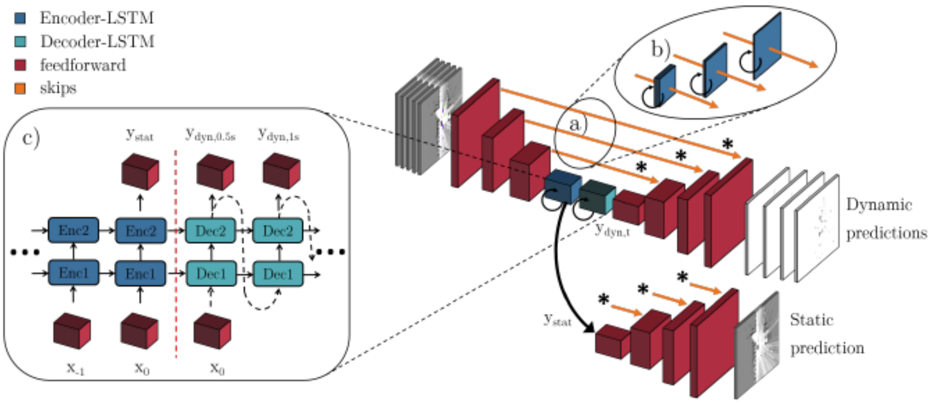


Figure 2.9: Schreiber, Hoermann, and Dietmayer [57]'s ConvLSTM encoder-decoder network with ConvLSTM skip connections.

difference learning architecture. The suggested method learns the difference between consecutive OGMs using a MFE algorithm. The output of the MFE algorithm is a tensor of the same height and width as the OGMs, which contains movement information in X and Y directions for each grid cell. This information is encoded and added to the encoded representation of the input OGM in the network's architecture. Subsequently, the encoded OGM and movement information flows through a ConvLSTM core, after which it is decoded and results in the predicted change of motion for each grid cell of the OGM. This predicted change of motion is added to the input OGM and processed by a feed-forward classifier layer that provides the prediction of the full OGM for one time step. The predicted OGM is inserted back into the network to predict the next time step. This is iterated for the number of predicted time steps that are desired. The network is trained on the KITTI [22] dataset. The difference learning architecture outperforms [15]'s architecture in predicting future OGMs.

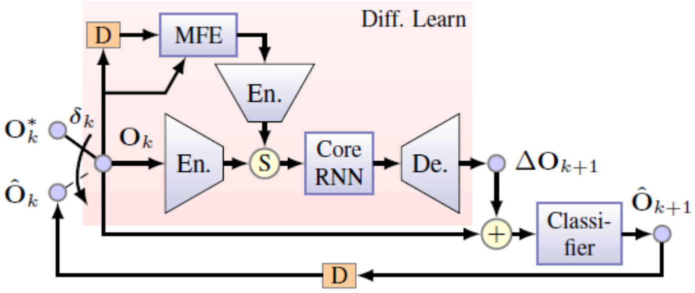


Figure 2.10: Mohajerin and Rohani [44]'s ConvLSTM-based difference learning OGM prediction network architecture. Two consecutive OGMs go through the MFE module to obtain a movement information for each grid cell. The encoded movement information is added to the current encoded OGM and processed by the Core RNN (the ConvLSTM). After decoding the result, it is processed by the Classifier to predict an OGM for one time step. To predict the next time step, the predicted OGM is inserted as the new input of the network.

2.3.4. PredNet-based OGM predictors

This subsection covers three methods that are all based on Lotter, Kreiman, and Cox [39]'s Predictive Coding Network (PredNet) architecture, which implements ConvLSTM modules (as is shown in figure 2.11). First, [27] adjusts PredNet to be suitable for OGM prediction. Second, [35] attempts to improve [27]'s architecture by adding an attention mechanism that better learns dependencies between sequential grid cells. Third, [61] extends [27]'s architecture to separate the static from the dynamic environment so the network can better distinguish dynamic behavior.

Initially, Itkina, Driggs-Campbell, and Kochenderfer [27] propose to utilize PredNet, an architecture based on the human brain predictive coding principle. This principle hypothesizes that the human brain continually predicts its incoming stimuli (e.g. visual stimuli from the eyes), compares these predictions against the actual stimuli, and generates an error signal to update the predictions. To mimic this principle, the PredNet architecture consists of multiple concatenated modules that all contain the same four parts: an input layer, a representation layer, a prediction layer, and an error representation layer (see figure 2.11). The input layer obtains features from the OGM (i.e. the actual stimulus) using convolutions and pooling. Then, a prediction of those features is made by the prediction layer (analogous to the brain's predictions), using the representation layer's features. The error representation layer generates an error by subtracting the input features from the predicted features (like the brain's generated error signal). After non-linearizing, the error is linked to both the representation layer and the module's output. The representation layer learns temporal and spatial patterns from the error and the representation layers of neighboring modules using a ConvLSTM [67]. Recurrently, the prediction layer then uses the representation layer's features to make predictions of the input features. Originally, the ConvLSTM was used for video prediction, which is a similar task to OGM prediction due to the similarity of images (frames in a video) and OGMs. Video prediction methods that are relevant for OGM prediction are further described in section 2.4.

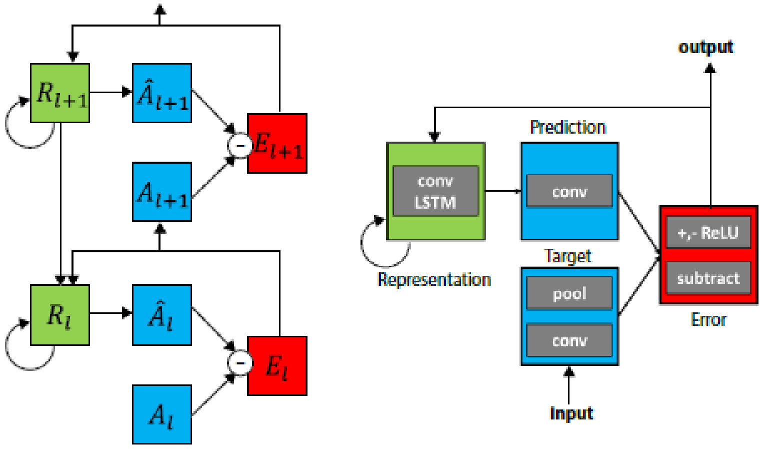


Figure 2.11: "Predictive Coding Network (PredNet). Left: Illustration of information flow within two layers. Each layer consists of representation neurons (R_l), which output a layer-specific prediction at each time step ($\hat{A}_l$), which is compared against a target (A_l) to produce an error term (E_l), which is then propagated laterally and vertically in the network. Right: Module operations for the case of OGM sequences." [39].

Itkina, Driggs-Campbell, and Kochenderfer [27] re-purpose the ConvLSTM [67] for OGM prediction. To train the PredNet, [27] generates OGMs from the KITTI [22] dataset. Itkina, Driggs-Campbell, and Kochenderfer [27] compares the PredNet's performance with a baseline that assumes a static environment for the short period of time that the predictions last, with an Fully Convolutional Network (FCN) network, and with a particle filter predictor. The research of [27] found that the PredNet outperforms the other investigated methods. However, for longer predictions, the objects in the OGM become blurry or even disappear from the environment.

To counter the blurriness and object disappearance for longer-term predictions, [35] proposes the AAConvLSTM for environment prediction. Lange, Itkina, and Kochenderfer [35] proposes to reduce blurriness by implementing this attention mechanism, which originated from creating long-term dependencies in language processing, into the PredNet architecture. Attention augmented convolutions replace the regular convolutions of the original ConvLSTM to form the AAConvLSTM. These attention augmented convolutions highlight inter-dependencies between spatial and temporal dimensions of the OGM sequences, allowing the network to learn object representations. Using attention augmented convolutions, compared to regular convolutions, also allows the LSTM to store more relevant information in its long-term memory. This improves the network's prediction accuracy. Lange, Itkina, and Kochenderfer [35] trains the AAConvLSTM PredNet using the Waymo [60] and KITTI [22] datasets. The results are compared with results from the original PredNet architecture and the PredRNN++ [62] architecture, which is also used in the video prediction domain. The AAConvLSTM performs better than the other investigated methods. However, blurriness and disappearance of objects remain as can be seen in figure 2.12.

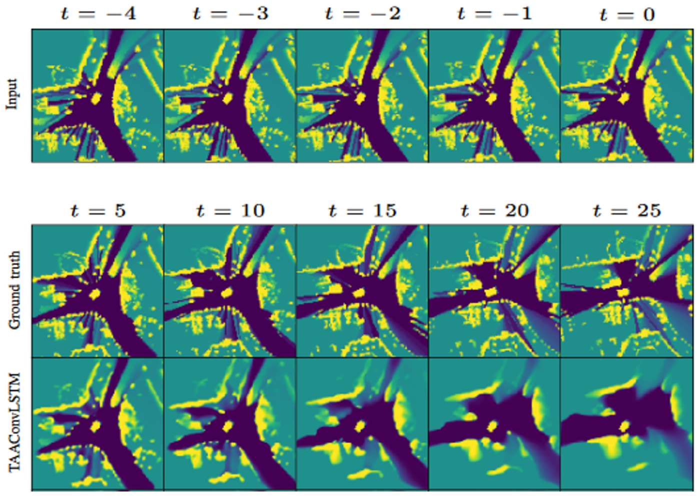


Figure 2.12: The AAConvLSTM predictions [35] (bottom) compared to the ground truth (middle) for 25 time steps, given the past input sequence of 5 frames (top). The predictions become increasingly blurry for the longer time horizons and objects disappear.

Toyungyernsub et al. [61] also aims to counter the blurriness and disappearing objects that result from [27]'s method for long-term predictions. Toyungyernsub et al. [61] expects that incorporating dynamic information directly into the network's architecture can solve the problems of [27]'s network. Therefore, they propose a double-prong network based on the PredNet architecture (figure 2.13 shows the network's pipeline). This network splits its architecture into a static and a dynamic prong. The static prong takes as input the static OGMs and makes static predictions, while the dynamic prong does the same for DOGMas. Toyungyernsub et al. [61] uses object detection and tracking to determine each grid cell's velocity between two separate OGM frames. Subsequently, the static and dynamic OGM predictions are fused to form a joint prediction. The double-prong network is trained on the Waymo [60] dataset. The method outperforms the baselines, however, the blurriness and disappearance of objects still remain for longer-term predictions.

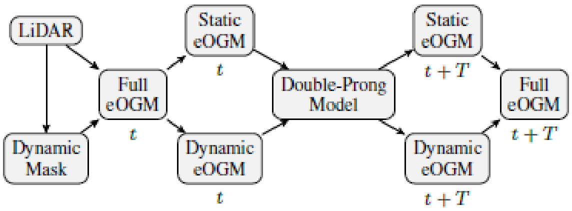


Figure 2.13: The Double-Prong network pipeline proposed by Toyungyernsub et al. [61]. This network separately predicts the future static and dynamic OGMs and then fuses them to create the complete OGM prediction.

Method	Dataset	Network	Input	Output	Loss Function	Metric
[15]	Own	RNN (GRU)	OGM +S*	OGM +S*	Cross-entropy	F1-score
[25]	Own	CNN	DOGMa	OGM	MSE	ROC, TPR, FPR
[57]	Own	ConvLSTM	DOGMa	OGM	L1-Loss, MSE	ROC, F1-score
[44]	KITTI [22]	ConvLSTM variations	OGM	OGM	Cross-entropy, MSE, SSIM [64]	TPR, TNR, SSIM [64]
[27]	KITTI [22]	PredNet	DOGMa, OGM	OGM	L1-Loss	MSE
[35]	KITTI [22] Waymo [60]	PredNet AAConvLSTM	OGM	OGM	L1-Loss	MSE, IS [4]
[61]	Waymo [60]	PredNet Double-Prong	DOGMa, OGM	OGM	L1-Loss	MSE, IS [4]
[66]	NuScenes [7]	MotionNet	OGM/BEV maps	DOGMa +S*	L1-Loss, Cross-entropy, consistency losses	MSE, MeSE

Table 2.2: An overview of properties of the different object-agnostic prediction methods. * +S stands for additional semantic data.

2.4. Video Prediction

OGM prediction methods such as [35]'s PredNet based [39] AAConvLSTM originate from video prediction. Since an image is similar to an OGM, a sequence of image frames i.e. video is similar to predicting a sequence of OGMs. Therefore, this section highlights relevant video prediction methods that are also the foundation of many OGM prediction methods.

The influential work by Xingjian et al. [67] introduced the Convolutional Long Short-Term Memory (ConvLSTM) network as an extension of the Convolutional Neural Network (CNN) to process sequences of images. Initially developed for precipitation nowcasting, the ConvLSTM uses past radar maps to predict a sequence of future maps. This model outperformed optical flow algorithms and the Long Short-Term Memory (LSTM) [24] due to its ability to capture complex spatiotemporal patterns. However, it also produces increasingly blurry predictions for greater prediction horizons. Finn, Goodfellow, and Levine [19] use ConvLSTM layers in their video prediction model, Convolutional Dynamic Neural Advection (CDNA). This model predicts pixel motions of frame segments and merges the segments into a single future frame prediction. Predictions made by CDNA also become blurrier over time due to the use of MSE loss in training, which causes uncertainty to be encoded as blur [19], as was also demonstrated in [35]'s OGM prediction method. To counter the blurriness, [19] suggest using an alternative loss. Subsequently, PredRNN++, the network [62] proposed, mitigates the blurring of predictions by employing a 'gradient highway' that provides shorter routes for gradients to flow through the network. This allows the network to learn stronger spatial correlations and short-term dynamics, leading to more confident predictions and decreased blurriness over long prediction horizons. Furthermore, the PredNet [39] network, as discussed in section 2.3.4 has been used for OGM prediction [27, 35, 61] because of its desirable properties. Based on the neuroscientific concept of predictive coding, each layer of PredNet makes a local prediction using ConvLSTMs and forwards the errors of that prediction to the subsequent layers. The authors argue this ensures the network learns an implicit model of the objects in the scenes, including their movement.

These approaches demonstrate that blurriness is partly caused by the model's increasing uncertainty with time. This uncertainty could be reduced by ensuring more informative supervision signals are used in training through better loss functions or by improving the ability of a model to recognize dynamic objects and their movement patterns.

2.5. Loss functions used in OGM prediction

Table 2.2 shows an overview of the object-agnostic OGM prediction methods. In the column 'Loss function', the loss functions used in each method to train the network are listed. The majority of the methods use per-pixel losses such as the L1 loss [27, 35, 61, 57, 66], the MSE loss [25, 57, 44], and the Cross-Entropy loss [15, 44, 66]. As is covered in section 1.4.2, these per-pixel losses do not measure any relations between grid cells. Therefore these losses cannot evaluate the distance errors nor the existence of objects. As a result, the loss function does not allow the network to recognize spatial relations between environment features causing the network to represent uncertainty as a blur. Mohajerin and Rohani [44], implements the inter-pixel SSIM loss [64] in combination with the MSE and Cross-Entropy loss to train their network. Their results do outperform the state-of-the-art alternatives. However, they do not investigate the isolated effect of using the SSIM loss, so its effect on the predictions is uncertain. Wu, Chen, and Metaxas [66] also implements their designed spatial and temporal consistency losses together with the L1 and Cross-Entropy loss. The consistency losses are based on the assumptions that an object's grid cells move equally and with minimal dynamic changes per time step and that the background remains static. The first two are the spatial and foreground temporal consistency losses. These losses minimize the motion between the grid cells corresponding to a single object and the average motion of a single object's grid cells through time, respectively. The third loss is the background temporal consistency loss and minimizes the movement of grid cells that do not belong to any annotated objects in the predictions. The three losses use semantic information to evaluate objects (i.e. clusters of grid cells between which dependency is assumed). Therefore, these losses do consider grid cell dependencies within detected objects. The downside is that relations between objects are still not considered in the loss functions and that these losses still rely on some object identification method.

2.6. Occupancy Grid Map semantic segmentation

As explained in section 1.4.3, OGMs can be extended with channels containing semantic information for each grid cell. Toyungyernsub et al. [61] show that adding semantic information to the OGMs, by distinguishing between static and dynamic objects, improves the accuracy of their network's predictions by decreasing the amount of blur and number of disappeared objects. Dequaire et al. [15] also implements a semantic segmentation functionality in their network. They train the network to detect and track objects and to estimate the future occupancy and corresponding semantic classes for originally occluded regions, which is shown in figure 2.14. They use multi-task learning to improve the semantic segmentation task. However, both approaches depend on semantic priors generated by an identification mechanism and are thus still dependent on the accuracy of this additional mechanism.

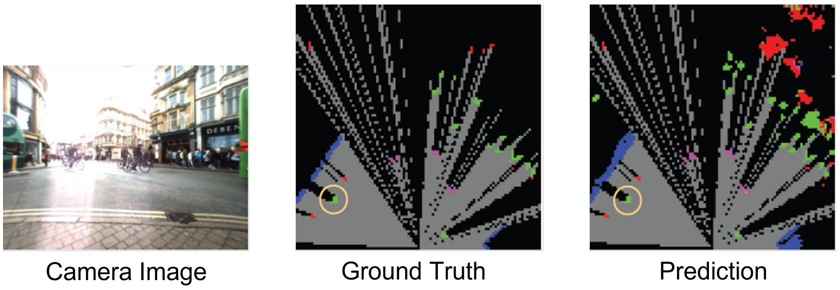


Figure 2.14: The semantic segmentation prediction from [15]'s network. The OGMs represent the scene from the camera image (left), with the AV's point of view at the bottom of each OGM. The ground truth is shown in the center and the prediction on the right. The network learns to fill in the originally occluded spaces, including their semantic classes. Black: occluded, Grey: visible, Blue: vehicle, Green: pedestrian, magenta: cyclist, red: static.

Wu, Chen, and Metaxas [66] avoids this issue by estimating the semantic state of a predicted future representation of a scene without relying on semantic priors. Their proposed model, MotionNet, has three different output heads that learn to predict channels with occupancy, semantic, or dynamic information for each grid cell. MotionNet extracts information from proposed regions of interest in the input and derives dynamic and semantic properties from those regions to predict the future semantic classes and dynamic states. However, the effects of having three different output heads, of which two heads predict semantic classes and dynamic states, respectively, on the performance of the occupancy information head (multi-task learning) are not investigated.

2.7. Multi-task learning for Occupancy Grid Map prediction

As mentioned in the previous section, [15] uses multi-task learning by simultaneously training their network to perform the object tracking and OGM prediction task and the semantic segmentation task. They posit that the network would learn hidden representations of objects from the tracking and prediction task, which it could use to improve the semantic segmentation task. According to the description in section 1.4.4, [15]'s source task is to learn object tracking and OGM prediction, while their target task is to perform semantic segmentation. Furthermore, [58] proposes an end-to-end multi-task network that predicts current-time DOGMas, including semantic labeling of objects, given 3D LiDAR data of the environment. Their network simultaneously learns a sensor model task, which converts 3D LiDAR data into OGMs, and the dynamic estimation and semantic segmentation tasks. They show that learning the source task of converting LiDAR data into OGMs improves the network's target task of semantic segmentation.

The second contribution of this thesis is to investigate whether the principle of multi-task learning, used by [15] and [58], can also be applied to improve a network's OGM predictions, given that it is also trained to perform semantic segmentation of the OGMs. In this method, hidden representations learned from the semantic segmentation source task are expected to be used to augment the OGM prediction target task. This would allow the network to identify dynamic objects and reduce the blurring of objects in long-term predictions without direct reliance on an object detection pipeline for the object predictions.

3 Method

OGM prediction is a self-supervised sequence-to-sequence prediction problem in which the goal is to predict a sequence of future OGMs given a sequence of past ones. Either binary OGMs or evidential OGMs [47] are used, as described in section 1.4.1. However, the method can be generalized to other OGM types. An OGM of the environment $X_t \in \mathbb{R}_+^{H \times W \times 1}$ at time step t contains the occupancy (i.e. whether it is free or occupied) probability of each grid cell $x_{t,ij},\ i \in [1, H],\ j \in [1, W]$ at that instant. Given a past (observed) sequence of OGMs $X_{-\tau:0}$, the objective is to predict T frames into the future $X_{1:T}$. This thesis proposes a deep learning method to perform these predictions. The prediction network is described in the following section. Afterwards, the methods to perform the loss function experiments, followed by the method for the semantic segmentation experiment, are discussed. Finally, the performance metrics to evaluate the quality of the predictions are highlighted.

3.1. Prediction network

This thesis employs the PredRNN++ [62] as the prediction network because it is a state-of-the-art sequence-to-sequence network. Originating from the video prediction domain, PredRNN++ has been used by [35] to predict OGMs. The PredRNN++ is designed to handle deep recurrence depths by employing a Causal LSTM, and back-propagation for long-term modeling by using a Gradient Highway Unit (GHU). The Causal LSTM is a structure that extends the normal LSTM and increases the recurrence depth from one time step to the next. It does so by using more non-linear layers than the normal LSTM. Furthermore, the causal LSTM contains a dual memory system consisting of a temporal memory and a spatial memory for each time step and hidden layer in the network. Through a cascaded mechanism, the spatial memory is a function of the temporal memory via a set of gate structures. This endows the output with a larger receptive field of the input at every predicted time step. These measures allow a more powerful modeling capability of spatial and temporal correlations. The GHU solves gradient back-propagation issues, such as a vanishing gradient, for long-term modeling by constructing a gradient highway. This highway forms a short route from the future outputs to the inputs through the network. The final architecture of the PredRNN++ is created by stacking L Causal LSTMs and inserting a GHU between the first and second Causal LSTM. The Causal LSTM and GHU complement each other by capturing both long-term and short-term dependencies in the input sequences. Figure 3.1 shows a schematic of the PredRNN++ [62] architecture.

3.2. Loss functions

Two inter-pixel loss functions, the Structural Similarity Index Measure (SSIM) [64] loss and the Sinkhorn loss [30, 13] are proposed to use for training OGM prediction networks. The expectation is that using an inter-pixel loss function improves the prediction quality compared to a per-pixel loss function. The L1 loss is considered the baseline loss because it is a simple, popularly used per-pixel loss. First, this thesis proposes to investigate the properties of the SSIM and Sinkhorn losses and compare them to the L1 loss prior to implementing the losses for OGM prediction on a real dataset. These investigations are described in the toy experiments in section 4.1. Then, the effects of training the PredRNN++ [62]

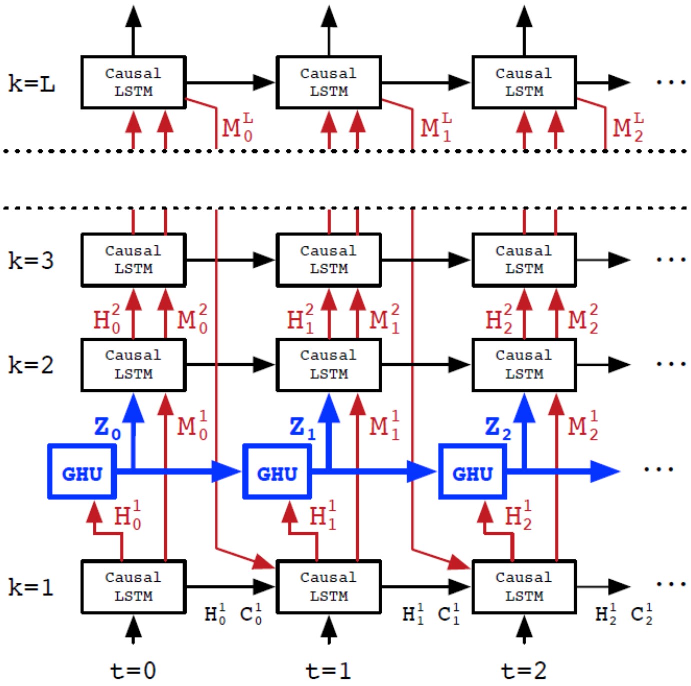


Figure 3.1: The PredRNN++ [62] architecture consists of L layers of Causal LSTMs with a GHU layer in between the first and second Causal LSTM layer. The spatial memory M_t^k flows through the layers. From the deepest layer the spatial memory cascades into the next time step as a function of the temporal memory C_t^k, which flows through the time steps. The transition features between the first and second layer, H_k^1, first flow through the GHU which connects all temporal layers by hidden state Z_k. This allows a short passage way for the gradient to flow through the network's inputs when back-propagating.

on a real large-scale dataset with different loss functions on the prediction quality are investigated. The L1, SSIM, and Sinkhorn losses are explained in the following subsections.

3.2.1. L1 loss

The L1 loss takes the mean of the absolute error between the grid cells of the ground truth X and prediction $\hat{X}$, as shown in equation 3.1.

$$\text{L1}(X, \hat{X}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} |x_{ij} - \hat{x}_{ij}| \tag{3.1}$$

3.2.2. SSIM loss

The Structural Similarity Index Measure (SSIM) [64] is a function inspired by the human visual system that measures the similarity between two images. It evaluates luminance, contrast, and structural information over windows of an image to estimate perceptual visual similarities [64]. Hence, the final loss value is computed by sliding the window over the image, calculating the SSIM locally, and taking the mean of the resultant values. The window size is a parameter that can be changed to adjust the scale for local evaluation. The components of the SSIM function are described in this section in the context of its application to OGM prediction.

Let the SSIM windows have size $N \times M$. The luminance μ_x of a window x of an OGM is the mean intensity over the window, which is calculated as in equation 3.2. The luminance values of corresponding windows y and x of the ground truth and prediction, respectively, are compared using equation 3.3, where the 'stabilising' constant $C_1 = (K_1 L)^2$, L is the dynamic range of the pixel intensity values, and $K_1 \ll 1$ is a small constant.

$$\mu_x = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij} \tag{3.2}$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{3.3}$$

The contrast is the standard deviation σ_x of the window, shown in equation 3.4. The contrasts of the windows of the ground truth and prediction are compared using equation 3.5, where 'stabilising' constant $C_2 = (K_2L)^2$, L is the dynamic range of the pixel intensity values and $K_2 \ll 1$ is a small constant again.

$$\sigma_x = \left(\frac{1}{NM-1}\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{ij}-\mu_x)^2\right)^{\frac{1}{2}} \tag{3.4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{3.5}$$

Equation 3.6 is used to compare the structures, or correlations, between the prediction and the ground truth, where C_3 is a small constant and σ_{xy} is computed using equation 3.7.

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{3.6}$$

$$\sigma_{xy} = \frac{1}{NM-1}\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{ij}-\mu_x)(y_{ij}-\mu_y) \tag{3.7}$$

After the comparisons are performed, the SSIM combines them according to equation 3.8. $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters to adjust the relative importance of the three factors. The SSIM ranges from -1 to 1, where a value of 1 means that the evaluated predicted window is identical to the ground truth one.

$$\text{SSIM}(x,y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma \tag{3.8}$$

The SSIM should be suitable for an OGM prediction loss because the luminance, contrast, and structural factors would optimize for equal occupancy values, equally 'sharp' object borders, and equal inter-object distances respectively between the ground truth and predictions.

3.2.3. Sinkhorn loss

The Sinkhorn loss [13] is an approximation of the Wasserstein distance [30] (or Kantorovich-Rubinstein distance). The Wasserstein distance expresses the 'distance' between two marginal probability distributions u, v as the minimum cost to transform one distribution to the other. Concretely, the Wasserstein-p distance for discrete distributions can be calculated using equation 3.9, following the formulation by Feydy [18].

$$W_p(u,v) = \left(\min_{\pi\in\Gamma(u,v)}\sum_{x,y} C(x,y)\pi(x,y)\right)^{\frac{1}{p}} \tag{3.9}$$

In the case of 2D marginal distributions, $\Gamma(u,v)$ is the set of all possible 4D distributions, or 'couplings', $\pi(x,y)$ that have marginals u and v. A coupling $\pi(x,y)$ expresses the probability mass that must be transported from position x to position y in order for the marginal distributions to be identical. The couplings are weighted by the cost function $C(x,y)$ for moving a unit of mass from x to y; the Wasserstein-p cost function family has $C(x,y) = \frac{1}{p}\|x-y\|_p^p$, with popular choices $p = 1$ (also called Earth Mover's Distance) and $p = 2$ (The L2 norm), where the latter is considered in this thesis. Optimizing the Wasserstein distance is computationally expensive, with higher-dimensional distributions incurring greater costs. To decrease the computational complexity, the Sinkhorn approximation [13] of the Wasserstein distance is used in this work instead. This approximation uses entropic regularization to ’blur’ the couplings to reduce the dimensionality of the optimization problem, as shown in equation 3.10.

$$S_\epsilon(u,v) = \min_{\pi \in \Gamma(u,v)} \langle C \odot \pi \rangle + \epsilon \mathrm{KL}(\pi, u \otimes v) \tag{3.10}$$

This equation consists of two terms: an objective and regularizing term. The objective term consists of the Hadamard product of the cost function C and the coupling π. The Kullback–Leibler (KL) divergence between the coupling and the Kronecker product of the marginals u and v is added to regularize the solution, weighted by a blur factor ϵ that balances the trade-off between accuracy and speed. To compute the Wasserstein distance for an OGM $X \in \mathbb{R}_+^{H \times W}$, the OGM is mapped to a discrete 2D probability distribution $P(X)$ by normalizing the cells by the total ’mass’ in the grid, shown in equation 3.11.

$$P(X) = \frac{X}{\sum_{i,j} x_{ij}} \tag{3.11}$$

Since the predicted and ground truth OGMs do not necessarily have the same mass, this mapping is a prerequisite to applying the Sinkhorn loss. The normalization may result in a re-weighting of correctly predicted ’local’ regions in the OGM, forcing the prediction network to be globally accurate to reduce the loss value on a local level. Furthermore, the Sinkhorn loss is expected to consider the spatial relations within OGMs because it optimizes the network to generate predictions that have the closest distance to the ground truth, when considering them as 2D distributions. Due to normalization, the prediction and ground truth will be evaluated without the original scaling information of each grid cell’s value. An expected downside of using the Sinkhorn loss is that the network will not learn the proper scaling of the OGMs, meaning that it might predict the correct locations for occupied and uncertain grid cells, but without the correct values (between 0 and 1). A proposed solution to this expected scaling problem is to add the L1 loss to the Sinkhorn loss and create a Sinkhorn-L1 composite loss. The L1 loss evaluates the absolute grid cell values and can complement the Sinkhorn loss with the correct scaling information. The Sinkhorn loss, in turn, implements the inter-pixel properties in the loss.

3.3. Semantic segmentation

The proposed method to perform semantic segmentation on the PredRNN++ network [62] is to provide the network with a sequence of evidential OGMs and predict a sequence of future evidential OGMs together with six semantic channels per future OGM. The six channels represent free space, vehicles, pedestrians, cyclists, static objects, and unknown space. Via the principle of multi-task learning, the hypothesis is that the network will improve its prediction performance when it also learns the task of semantic segmentation without providing semantic priors in the input. It is expected that the network will learn hidden representations of each semantic category and uses that information to generate more certain predictions (i.e. predictions with fewer disappearing objects and less blur). Moreover, by not including semantics in the input, the network will not be dependent on the object detection and tracking methods, unlike classic prediction methods.

3.4. Performance metrics

The results are evaluated using four metrics. The Mean Squared Error (MSE) is used because it is a popularly used per-pixel metric to evaluate OGMs (as in [27, 35, 61, 66]). To evaluate high-level inter-pixel similarities between the predictions and the ground truths, the Image Similarity (IS) [4] metric is used (as in [35, 61]). Furthermore, two metrics that are popular in the object detection domain [49, 52], the Average Precision (AP) and Accuracy metrics, are used as well. Respectively, the following sections elaborate on these metrics.

3.4.1. The Mean Squared Error

The MSE metric measures the per-grid cell difference between the ground truth and predicted OGMs. The formula is shown in equation 3.12, where Y is an $m \times n$ OGM, X its prediction, and i and j the OGM coordinates.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[Y(i,j) - X(i,j)]^2 \tag{3.12}$$

3.4.2. Image Similarity

IS is an inter-pixel metric that accounts for the difference in scene structure between the ground truth and predictions by computing a distance map for grid cells with the same occupancy value, then averaging over these values. Birk [3] developed the IS metric which they later used in [4]'s research on finding similarities between OGMs in order to merge them to generate bigger maps. [3] defines the IS metrics according to equations 3.13 and 3.14.

$$IS(m_1, m_2) = \sum_{c \in C} d(m_1, m_2, c) + d(m_2, m_1, c) \tag{3.13}$$

$$d(m_1, m_2, c) = \frac{\sum_{m_1[p_1]=c} min\{md(p_1, p_2)|m_2[p_2] = c\}}{\#_c(m_1)} \tag{3.14}$$

In the case of an evidential OGM, C denotes the occupancy values that are categorized as free (0), unknown (0.5) and occupied (1). $m_i[p]$ is the category c of grid m_i at position $p = (x, y)$. $md(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$ is the Manhattan-distance between p_1 and p_2. $\#_c(m_i) = \#p_1|m_i[p_1] = c$ is the number of pixels in m_i with category c [3]. The IS metric is the sum over all the grid's categories, of the average Manhattan-distance of the cells with category c (i.e. free, unknown or occupied) in grid map m_i to the nearest cell with category c in grid map m_j ($d(m_1, m_2, c)$), plus the average Manhattan-distance vice versa [3]. This metric penalises blurriness in predictions in this setup, since blurred (predicted) cells (i.e. with value around 0.5) will belong to a different category than the corresponding ground truth cell and warp the computed distance maps as a result. The higher the IS value, the larger the difference between the evaluated image and the ground truth.

3.4.3. Average Precision and Accuracy

The Average Precision (AP) and Accuracy are both metrics that measure individual OGM grid cell value correspondences between the prediction and ground truth. The AP measures the area under the Precision-Recall (PR) curve (see equation 3.15. Precision is the probability that a predicted positive grid cell is correctly predicted (see equation 3.16. Recall is the probability that a positive grid cell in the ground truth is also predicted as positive (see equation 3.17). The PR curve is generated by computing the Precision-Recall pairs for different probability thresholds. The Accuracy computes the fraction of correct predictions for each class as is shown in equation 3.18, where Y is an $m \times n$ ground truth OGM, X its prediction, and i and j the OGM coordinates. The OGMs consist of three classes, free, uncertain, and occupied, which are evaluated. The class values correspond to the values 0, 0.5, and 1 to which the OGM grid cells are rounded if their values range between $0 - 0.33$, $0.33 - 0.67$, and $0.67 - 1$ respectively.

$$AP = \sum_n (R_n - R_{n-1}) P_n \tag{3.15}$$

$$P = \frac{\sum TruePositive}{\sum PredictedConditionPositive} \tag{3.16}$$

$$R = \frac{\sum TruePositive}{\sum ConditionPositive} \tag{3.17}$$

$$Accuracy = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} 1(Y(i,j) = X(i,j)) \tag{3.18}$$

4 Experiments

This section describes several experiments that are performed to answer the research questions from section 1.2. First, four toy experiments are performed that compare the effects of using the per-pixel L1 loss with the inter-pixel SSIM, and Sinkhorn loss functions. These toy experiments, including their goals, methods, results, and conclusions, are enveloped in this experiments section because the findings are helpful for the comprehension of the rest of the experiments and results. Second, following up on the toy experiments, the OGM prediction experiment is performed in which the PredRNN++ [62] is trained using the different losses from the toy experiments and two additional losses that consist of a composite of the Sinkhorn and L1 losses. The effects of using these different loss functions are measured. Lastly, the multi-task learning experiment is performed in which the PredRNN++ network is trained to predict semantic segmentation channels together with the sequence-to-sequence OGM prediction task. It is measured whether learning the semantic segmentation task has an effect on performing the OGM prediction task.

4.1. Toy Experiments

First, four toy experiments are performed to demonstrate and investigate the effects of the L1, SSIM, and Sinkhorn loss functions before the main OGM prediction experiments are performed. The first toy experiment is the 'sliding square experiment'. The loss functions' distance capturing properties are investigated by computing the losses for an increasing distance error by sliding a square over an increasing distance compared to a ground truth square. This experiment can verify whether the proposed inter-pixel losses have a better ability to capture spatial relations between grid cells than the per-pixel L1 loss. Suppose the inter-pixel losses are able to capture spatial relations better than the L1 loss. In that case, this thesis hypothesizes that using these inter-pixel losses to train an OGM network will allow the network to learn better object representations and in turn will reduce blurriness and object disappearance. The second toy experiment investigates whether the inter-pixel losses are as effective to train a network compared to the per-pixel L1 loss. The L1 and Sinkhorn losses are compared when optimizing a simple 1D deep learning task. The SSIM loss is excluded because it can not perform on 1D tasks. The task for the network is to convert a 1D probability density function to a cumulative distribution function. The networks trained by the two losses are compared for increasing sizes of the 1D distribution vector to evaluate the performance consistency for increasing network sizes. The third toy experiment investigates whether the inter-pixel losses can also effectively train a 2D prediction network since the eventual task in this thesis is to train an OGM prediction network. The L1, SSIM, and Sinkhorn losses' ability to optimize a network for a simpler 2D image prediction task are evaluated. This task is to convert a random noisy image to a 2D Gaussian image, hereinafter referred to as the noise-to-Gaussian conversion task. After the different losses are investigated on their abilities to capture distances and train both 1D and 2D networks, their performance on training an OGM prediction network is investigated in the fourth toy experiment. However, before training on a real dataset, the loss performances on a controlled synthetic toy dataset are first investigated. In this toy dataset, there is only one small dynamic object (a pedestrian crossing the street), whereas the other objects in the scene are static (other vehicles and pedestrians). In case the inter-pixel losses are better at capturing distances and performing the 1D and 2D prediction tasks compared to the L1 loss, it is expected that those losses would better predict the future locations of the small dynamic pedestrian when the synthetic toy dataset is used for sequence-to-sequence OGM prediction. The four experiments, together with the results, discussions and conclusions, are described in this section.

4.1.1. Sliding square experiment

Goal To investigate and compare the distance capturing properties of the L1, SSIM [64], and Sinkhorn [13] losses on images. Also, compare the Sinkhorn loss to the Wasserstein [30] metric's behavior since the Sinkhorn loss is an approximation of the Wasserstein distance.

Method The distance capturing properties of the loss functions can be evaluated by designing two scenarios of ground truth and prediction images containing at least one object that gradually increases its distance (sliding) in the image compared to the ground truth. The values for each pixel in the image are either 0 (free) or 1 (occupied). By computing the loss over the ground truth and the predictions, the relation between the increasing distance error and the resulting loss values can be measured. The first scenario contains only one object to measure the distance error effects on the loss of an isolated object. The second scenario contains three objects, of which one will increase its distance error in the predictions, causing it to overlap another object in the process. The other two objects will always be predicted correctly. This is done to measure the influence of other, correctly predicted objects on the loss value.

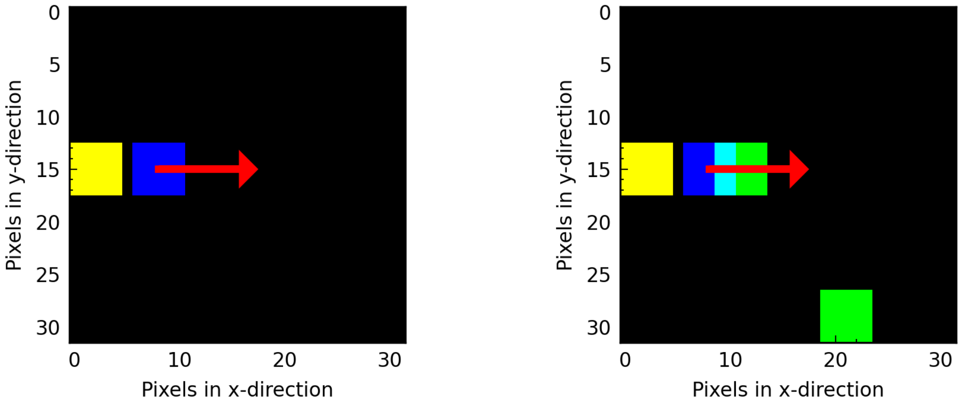
Experimental setup Two single-channel 32x32 ground truth images are generated, one for each scenario. The first image contains a single 5x5 filled square placed on the left side of the image. The second image contains the square from the first image and two additional 5x5 filled squares around the center and bottom of the image. Then, a set of simulated predictions is generated for both ground truth images. The simulated predictions contain a square (the left-most square in the ground truth) which is increasingly displaced to the right by 1 pixel (slides) for every prediction in the set until it reaches the right side of the image. Therefore, the prediction sets contain predictions with an increasing distance error for every additional frame. For the second scenario, the other squares are always predicted correctly. Figures 4.1a and 4.1b portray the first and second scenarios. Using the different loss functions (L1, SSIM, Sinkhorn) and the Wasserstein metric, the predictions are compared to the ground truths for both scenarios. For the SSIM loss, a window size of 11 is used. For the Sinkhorn loss, a blur of 0.1 is used, and both the Sinkhorn loss and Wasserstein metric use an L2 distance function. The experiment is performed using libraries compatible with the deep learning framework PyTorch [51]. For the losses and metric, the PyTorch L1 loss, the Torch Geometry library's [54] SSIM loss, the Geomloss library's [18] Sinkhorn loss, and the Python Optimal Transport library's [20] Wasserstein metric implementations are used. This experiment is repeated for different window sizes (7x7 and 9x9) and blur values (0.5 and 0.1) for the SSIM and Sinkhorn losses, respectively.

Evaluation method The resulting normalized loss/metric values are plotted versus the increasing distance errors to show the behavior of each loss and the Wasserstein metric. By visually analyzing the graphs, the behaviour of each loss/metric can be evaluated.

Results Figure 4.1 shows the two experiment scenarios. Figure 4.2 shows the normalized graphs for the L1, SSIM, and Sinkhorn losses, and the Wasserstein metric. The results of the additional hyperparameter experiments are shown in Appendix A.

Discussion The results from the first scenario in figure 4.2a show different behaviors for each loss and the Wasserstein metric. Starting with the losses from the first scenario, all show an increasing loss value while the moving square from the prediction overlaps the left ground truth square, after which the behaviors differ significantly.

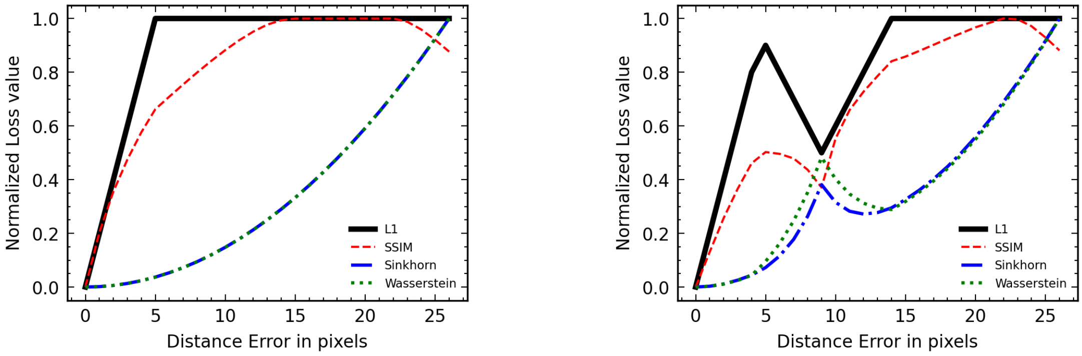
The L1 loss shows a linear increase until the prediction square no longer overlaps the ground truth square. From there, the L1 plateaus at the same value. This can be explained by the per-pixel evaluation property of the L1 loss. While the prediction overlaps the ground truth, the overlapping regions contain the same pixel values, which the L1 loss considers as correct predictions. As the prediction



(a) The first simulated scenario.

(b) The second simulated scenario.

Figure 4.1: The two scenarios that are generated to investigate the effect of the different loss functions on an increasing distance error. Yellow: Ground truth / False negative prediction, Green: True positive prediction, Blue: The moving square in the simulated predictions, Cyan: Overlap of the moving square and true positive predictions, Red arrow: The direction in which the moving square slides, Black: Free.



(a) The results for the first scenario.

(b) The results for the second scenario.

Figure 4.2: The normalized graphs for the L1, SSIM, and Sinkhorn losses, and Wasserstein metric for the two scenarios where the ground truths are compared to predictions with increasing distance errors.

square moves linearly, the overlapping region decreases linearly, while the non-overlapping regions increase linearly, until the prediction square is entirely separated from the ground truth. This corresponds to the linearly increasing L1 loss until the squares do not overlap anymore. The number of falsely predicted pixels stays the same after the prediction and ground truth squares do not overlap anymore, so the loss stays the same even if the distance error increases. Therefore, the results demonstrate that the L1 loss does not capture these distance errors.

The SSIM loss increases with the increasing distance error, but the slope of the increase decreases, generating a flattening curve. Notably, the slope shows sudden decreases in inclination at distance errors from 5, 14, and 23 pixels, where the loss curve even starts to decrease from a distance error of 23 pixels. These sudden slope changes can be related to the window size of the SSIM loss and may be explained as follows. The change between the distance errors of 5 and 6 pixels may be caused by separating the prediction and ground truth squares. The SSIM 11x11 sized windows that go over the image in y-direction, spanning the pixels $0 - 11$ in x-direction, can fit the prediction and ground truth squares exactly with a 1-pixel gap in between when the distance error is 6 pixels. Since the SSIM loss measures similarities corresponding to variance, correlation and intensity, the gap may be of such a significant influence on that similarity that the SSIM loss's slope decreases for increasing distance errors. At the distance errors between 14 and 15 pixels, the ground truth and prediction square cannot fit within one window anymore, even partially. This would mean that any distances that could be captured at first cannot be captured anymore due to the window size limitations, resulting in a plateauing value of the SSIM loss. Finally, at the distance errors between 23 and 24 pixels, the slope of the SSIM loss decreases. This effect may be caused by the lack of symmetry in the evaluation. As the predicted square closes in on the right edge of the image, the square will be considered less in the evaluations since fewer windows will cover the square due to the edge's limits. This may result in a decrease in the SSIM loss. A similar effect is seen for the window sizes of 7x7 and 9x9 in figure A.2 in Appendix A. For the Sinkhorn loss, the loss increases quadratically for an increasing distance error. This can be explained by the L2 distance cost implemented in the Sinkhorn loss. According to that cost, an increasing distance means an increased effort to change the prediction into the ground truth, resulting in the desired distance capturing behavior. The Wasserstein metric shows the same behavior as the Sinkhorn approximation. Figure A.3 in Appendix A shows that the blur value does not change the Sinkhorn's behavior for this scenario.

The second scenario shows similar results for each loss as the first scenario. Interesting to note is that the L1 and SSIM losses decrease. In contrast, the Sinkhorn loss and Wasserstein metric increase with a greater slope when the prediction square overlaps its own predicted 'middle' square at 10 pixels x-direction. For the L1 loss and SSIM loss, more overlap means more pixels have the same value and the structures remain more 'similar', which explains the decrease in loss value. This behavior is also visible for the SSIM loss with different window sizes in figure A.2 in Appendix A, though the decrease during overlap is much smaller.

For the Sinkhorn loss, the overlap means that more 'effort' is required to redistribute the weights to form the ground truth from the prediction. During overlap, there are fewer occupied pixels in the prediction than in the ground truth. If there is no overlap, the number of occupied pixels in the prediction and the ground truth are equal. Since the Sinkhorn loss works on normalized images, the prediction's occupied regions get re-weighted when there is 'overlap'. Therefore, to redistribute the weights from the prediction to form the normalized ground truth, the prediction must relocate weights from the middle as well as from the right-most and bottom squares to compensate for the lack of occupied pixels. This requires the weights to travel a greater distance over the image than when there is no overlap between the squares. For training a deep learning network, this behavior can be desirable if objects disappear in the predictions. In that case, it can be hypothesized that the Sinkhorn loss will increase such that the network learns to 'regenerate' the disappeared object (i.e. the lack of occupied grid cells compared to the ground truth) to decrease the error peaks. One could view the Sinkhorn's property to penalize both the distance error and mismatch of occupied cells, compared to the ground truth, as more desirable to maintain correct object instances in predictions, than the L1's and SSIM's property where having fewer occupied grid cells, compared to the ground truth, is rewarded.

Further, the Wasserstein metric shows similar behavior to the Sinkhorn approximation, but during the overlap the Wasserstein metric increases more than the Sinkhorn loss. Figure A.3 in Appendix A shows that different blur values change the increase of the loss during overlap. The Wasserstein metric is basically the Sinkhorn loss, but without any blur, which can explain why the increase of the loss during overlap differs, like the Sinkhorn losses with different blur values.

Conclusion The L1 loss does not capture distance errors, while the SSIM and Sinkhorn losses do. For larger distance errors the SSIM loss is limited by the window size. If the distance error is greater than can be measured within a window, the SSIM loss cannot capture them anymore. The Sinkhorn loss increases quadratically with increasing distance error, which is the desired behavior when evaluating OGM. The Sinkhorn loss is similar in its distance capturing behavior to the Wasserstein metric.

4.1.2. Probability density to cumulative distribution conversion

Goal Compare the performance of L1 loss and the Sinkhorn loss [13] to train a network to convert a 1D probability density function (pdf) to its cumulative distribution function (cdf), for increasing data and network sizes. With this comparison, a better estimate can be made whether the Sinkhorn loss, compared to the L1 loss, can perform equally well on larger data and network sizes, such as 128x128 grid cell OGMs and the PredRNN++ network.

Method To investigate the performances of training this probability density function (pdf) to cumulative distribution function (cdf) conversion network, a one-layer fully-connected feed-forward network with as many nodes as elements (m) in the pdf input vector is used to perform this task. For each element in the input pdf vector, the corresponding element in the output cdf vector should be the input element summed to all the previous elements in the pdf vector. The one-layer network has sufficient parameters, because with bias it would have $2*m*m+2*m = 2m(m+1)$ parameters, where $m+\sum_{k=1}^{m} k$ parameters are required which are fewer than the network has for all m. For either loss function, the network will be trained. Also, to compare the ability of each loss function to train larger networks, training the network is repeated several times. Each time the size of the network and elements (bins) in the pdf and resulting cdf are doubled. The quality of the resulting cdfs are measured for each network to compare the effects of using the L1 loss with the Sinkhorn loss.

Experimental setup To generate the input pdfs N samples are taken from a Normal distribution with mean and standard deviation equal to $m/2$. The samples are put in a histogram of m bins, which is normalized. The ground truth cdfs are then generated from the pdfs by taking their cumulative sum followed by normalization, so the Sinkhorn loss can process them. A fully connected one-layer feedforward network with m inputs and m outputs is used in this experiment. The network is optimized using Stochastic Gradient Descent [56]. Eleven networks are trained to perform pdf to cdf conversion, each with a different histogram bin number m, which doubles for each following experiment starting at 8 and ending at 8192. The number of samples N scales with the histogram number m to make sure the bins are filled to resemble the Normal distribution. Training is done for 1000 epochs, 1 sample per epoch with a batch size of 1. The learning rate is set to $1e-2$. The experiment is repeated five times, of which the average result values are taken for evaluation. The experiment is performed using the deep learning framework PyTorch [51] and compatible libraries for the losses. The PyTorch L1 loss and the the Geomloss library's [18] Sinkhorn loss implementations are used. An Nvidia Tesla K80 GPU is used for training and evaluation.

Evaluation method Evaluation is done quantitatively by comparing the MSE metric between the predictions and ground truths with the increasing histogram bin number m. The MSE and histogram bin number are plotted in a logarithmic scale to interpret the relation between the doubling histogram numbers and the MSE metric.

Results Figure 4.3 shows the graph where the logarithm of the MSE metric is plotted against the logarithm of the number of histogram bins m for the L1 loss and Sinkhorn loss networks. The plot shows the average values of five experiments. Figure 4.4 shows the input pdf and the corresponding ground truth cdf and predicted cdfs by the L1 and Sinkhorn networks for the increasing histogram size m.

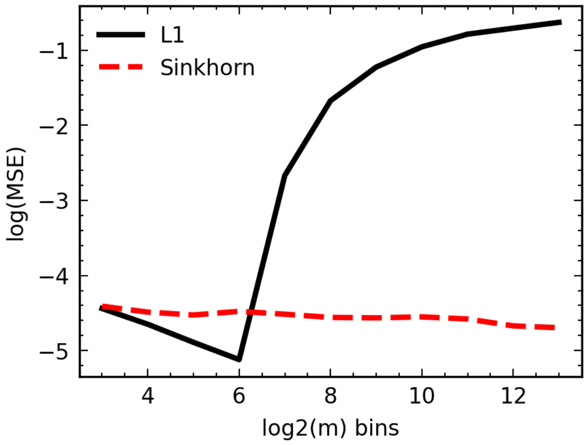
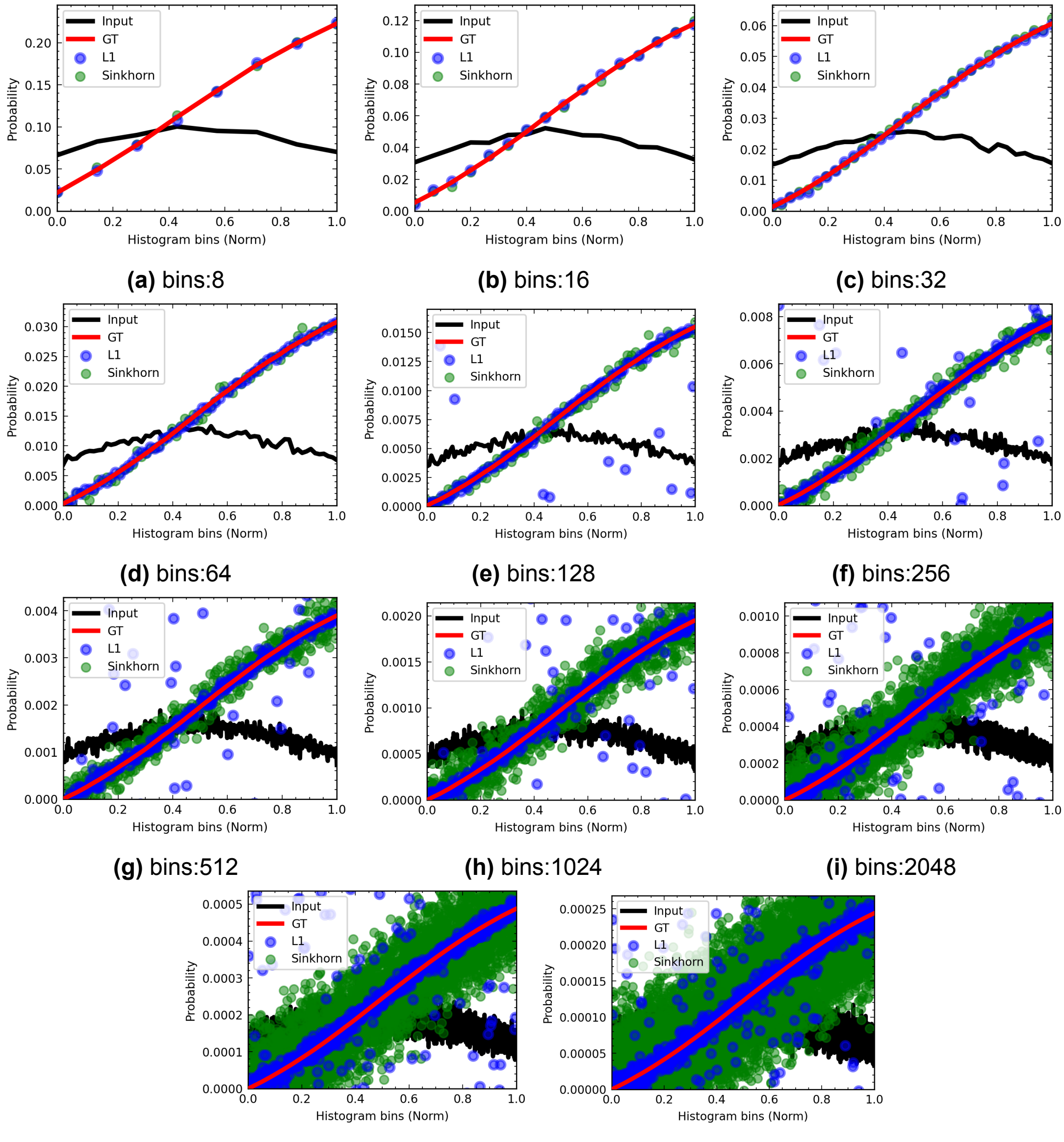


Figure 4.3: The log_{10} of the MSE metric versus the log_2 of the histogram bin number m. Black shows the L1 loss network performance. Red shows the Sinkhorn loss network performance.

Discussion Figure 4.3 shows that for an increasing number of histogram bins, the MSE for the L1 loss network first decreases, until the number of bins become larger than 64, after which the MSE increases. The MSE remains constant for the Sinkhorn loss network for an increasing number of bins.



(a) bins:8 **(b)** bins:16 **(c)** bins:32

(d) bins:64 **(e)** bins:128 **(f)** bins:256

(g) bins:512 **(h)** bins:1024 **(i)** bins:2048

(j) bins:4096 **(k)** bins:8192

Figure 4.4: The 1D pdf to cdf conversion results for increasing histogram bin number m. Each figure shows a result for a different network size with m nodes. The figures show the input pdf (black), the corresponding ground truth cdf (red), and the predictions from the L1 loss network (blue) and Sinkhorn loss network (green). Note that the y-axis values range changes with increasing bin number because the values are normalized with respect to an increasing number of values.

Both networks show similar performance (between 1e-4 and 1e-5) up to 64 bins, after which the L1 network diverges, with higher MSE values, from the Sinkhorn network performance. The graphs in figure 4.4 demonstrate that the predictions from both the L1 and Sinkhorn loss networks show similar, accurate results compared to the ground truth until the number of bins m is 128. From this point, the L1 loss network predictions have an increasing number of outliers, while the Sinkhorn's number of outliers seems consistent. On the other hand, the bulk of the L1 predictions is closer to the ground truth than the Sinkhorn predictions. The L1 loss' per-pixel property, by not considering dependencies between the different output values, could explain the increasing number of outliers. Each value is optimized individually causing some parts to 'lag behind', especially when the network size increases which is demonstrated by the increasing number of outliers. The Sinkhorn loss considers spatial relations, so optimizing one value also influences other values, which can explain the smoother convergence also for larger bin numbers. The larger, but constant variance by the Sinkhorn loss compared to the L1 loss could be explained by the blur parameter. Perhaps it is not able to converge the output with better precision because it is limited by the blur parameter.

Conclusion For an increasing number of histogram bins starting from 64 bins, the pdfs to cdfs conversion network performs better when trained by the Sinkhorn loss compared to the L1 loss according to the MSE metric. For smaller number of bins, the performance is almost equal, with the L1 loss network performing slightly better. The expectation is that the performance trend of the Sinkhorn loss is maintained for larger data formats, such as OGMs, using larger networks.

4.1.3. Noisy image to 2D Gaussian conversion

Goal Compare the performance of networks trained using the L1, SSIM [64], and Sinkhorn [13] losses to train a network to convert a 2D uniform noise image to a 2D Gaussian image (noise-to-Gaussian conversion). Through this comparison, a better estimate can be made whether the SSIM and Sinkhorn losses, compared to the L1 loss, can perform equally well on networks that process images (or data formats such as OGMs that are similar to images).

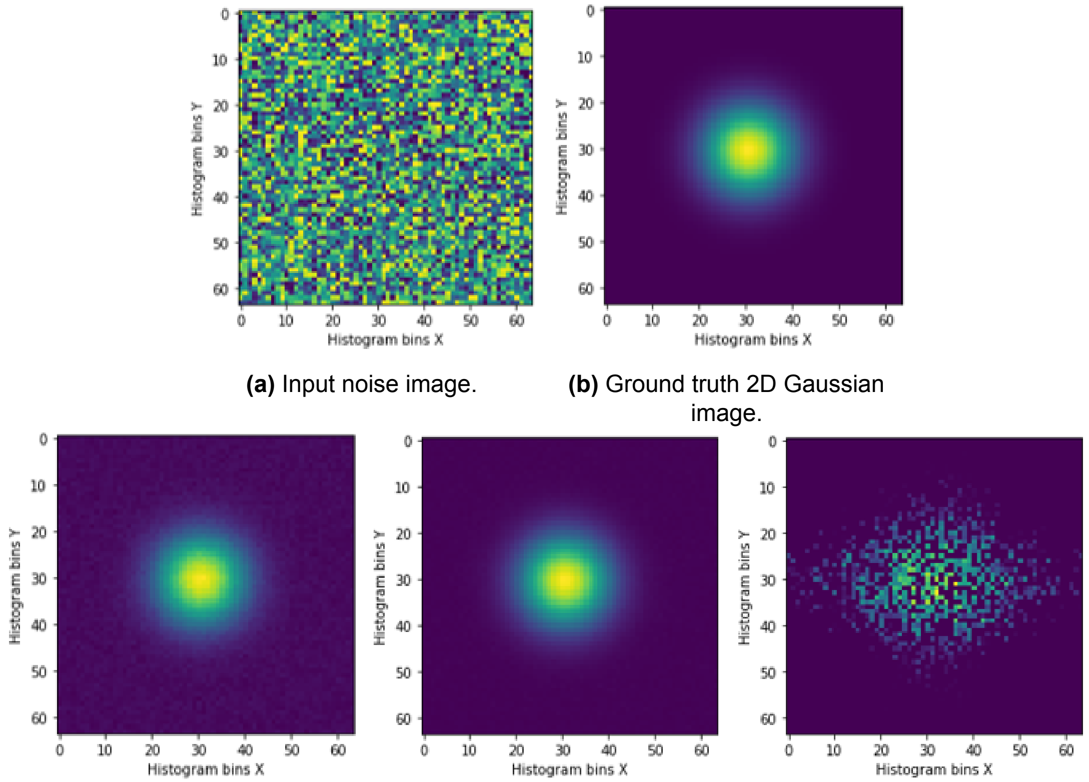
Method To investigate the performances of training this noise-to-Gaussian conversion network, a multi-layer fully-connected feed-forward with an activation function is created for this task. The L1, SSIM, and Sinkhorn losses are used for training, after which the results are compared to evaluate the effect of using each of the losses.

Experimental setup The inputs are randomly generated 64x64 uniform noise images. The ground truth is a 2D Gaussian distribution with a mean of 0 and standard deviation of 1, which is mapped to a 64x64 grid with the mean in the center (at $(32{,}32)$). Both the inputs and ground truth are normalized so they can be processed by the Sinkhorn loss. A simple deep learning network is designed for this task consisting of a linear layer, followed by a ReLU, followed by two linear layers. Training is done for 2000 epochs, 1 sample per epoch with a batch size of 1. The learning rate is set to 1e-3, 1e-4, 1e-2 for the L1, SSIM, and Sinkhorn losses respectively. The network is optimized using Stochastic Gradient Descent [56]. The experiment is performed using the deep learning framework PyTorch [51] and compatible libraries for the losses. The PyTorch L1 loss, the Torch Geometry library's [54] SSIM loss, and the the Geomloss library's [18] Sinkhorn loss implementations are used. An Nvidia Tesla K80 GPU is used for training and evaluation.

Evaluation method The network's resulting predictions of the 2D Gaussian image for each loss are evaluated quantitatively by using the Kullback-Leibler (KL) divergence metric since it can measure how much two distributions (the ground truth and predicted 2D Gaussians) are similar to one another. Also, a qualitative visual comparison is made.

Results Figure 4.5 shows the results from the noise-to-Gaussian conversion networks using different loss functions.

Discussion The results show that the SSIM result has the best KL divergence score, where the L1 result is double as high and the Sinkhorn result about 30 times as high. This can also be seen in the figures. Both the L1 and SSIM loss networks converge to generate 2D Gaussian images given a random noise input. The L1 loss network shows a light noise throughout the predicted image, while the SSIM shows a visually identical image compared to the ground truth. This could be explained by the SSIM's property of optimizing the output to resemble the ground truth visually. In contrast, the L1 loss optimizes for each pixel to have the minimal absolute error. So, having a better visually resembling prediction also has a better resembling distribution according to the KL scores. The Sinkhorn loss network generates predictions that show a scatter of lighter pixels that cluster around the center of the image, resembling a scattered version of the ground truth image. After blurring the Sinkhorn predictions and the ground truths they might be more similar, and thus, the network would not be able to converge further. Perhaps using a lower blur for the Sinkhorn loss would result in a better performance.



(a) Input noise image.

(b) Ground truth 2D Gaussian image.

(c) L1 prediction. KL: 0.0137

(d) SSIM prediction. KL: 0.00613

(e) Sinkhorn prediction. KL: 0.188

Figure 4.5: The visual results from the noise-to-2D-Gaussian image conversion networks using different loss functions together with the average KL divergence values for using that loss, where lower is better.

Conclusion When comparing the performances of the noise-to-2D-Gaussian conversion networks, the network trained using the SSIM loss performs best both quantitatively and qualitatively. The L1 loss network performs almost as well but predicts a noisier image than the SSIM loss network. The network trained using the Sinkhorn loss performs worst. All in all, all three losses show the capability of evaluating predicted images and updating the network that leads to convergence to the ground truth.

4.1.4. Synthetic dataset OGM prediction

Goal Compare the performance of a sequence-to-sequence OGM prediction network trained using the L1, SSIM [64], and Sinkhorn [13] losses on a synthetic dataset containing scenes with one small, dynamic pedestrian and static surrounding objects. By comparing the performances of the losses using a controlled dataset, it is expected that a better controlled evaluation can be made compared to using real data.

Method For this experiment, a similar method is used as is described in the method for the loss experiments of this thesis in section 3. This method also uses the PredRNN++ network and evaluates the results using the same metrics. This method, however, uses a synthetic toy dataset to train the OGM prediction network, which is described in the next paragraph. Besides the L1, SSIM, and Sinkhorn losses, the network is also trained on a Sinkhorn-L1 composite loss which is the sum of the Sinkhorn and L1 losses. This composite loss is investigated because it is expected that the network trained on the Sinkhorn loss will not learn the proper scaling of the OGMs. After all, the Sinkhorn loss is evaluated on normalized data.

Dataset For this toy experiment a synthetic dataset is generated consisting of 500 30-second 128x128 binary OGM sequences. Each sequence consists of a simulated pedestrian that crosses a scene with static vehicles and pedestrians. For each sequence, a random number of vehicles (rectangles of approximately 40x25 grid cells) and pedestrians (squares of approximately 4x4 grid cells) are generated and placed in an order that resembles a straight road with sidewalks. The vehicles would be placed vertically or horizontally in the center, depending on the road orientation. Pedestrians would be placed along the outer lines of the vehicles as if they are standing on a sidewalk. Besides these static traffic actors in the scene, a dynamic pedestrian is placed randomly on one of the outer 'sidewalks' and crosses the road filled with the vehicles. The RRT* [31] motion planning algorithm is used to plan the path for the pedestrian because it contains some randomness. Yet, it converges to an optimal, shortest path while avoiding the obstacles. The randomness of RRT* serves as a way to resemble the diversity of human walking styles. The 500 30-second sequences can subdivided into 1500 10-second sequences to train the network in the experiment.

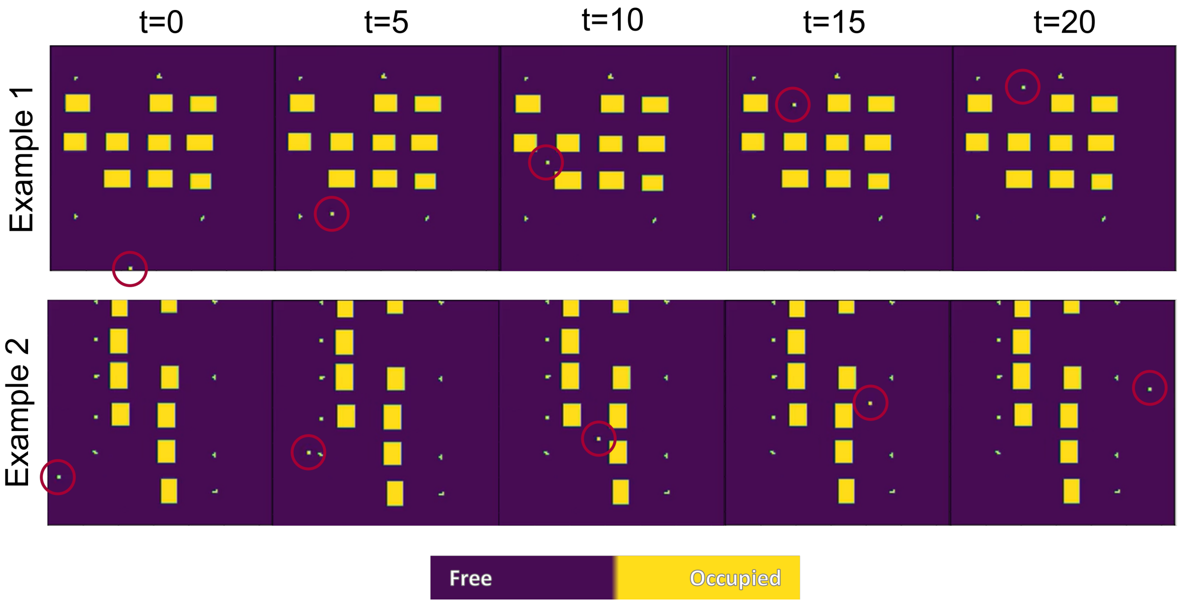


Figure 4.6: Two 20-frame example sequences with a five-second interval of the synthetic toy dataset. The binary OGMs of size 128x128 contain static vehicles (large rectangles), static pedestrians (small squares), and one dynamic pedestrian (encircled) crossing the scene. The number of vehicles and pedestrians, the dynamic pedestrian crossing direction, and road direction (horizontal or vertical) are randomized for each sequence.

Experimental setup This experiment evaluates the effects on the test-time performance of training the PredRNN++ OGM prediction network [62] with each of the L1, SSIM, Sinkhorn, and Sinkhorn-L1 composite losses. The OGM sequences from the synthetic dataset. A dataset of 300 training, 40 validation, and 100 testing non-overlapping sequences of 10 frames. Additionally, 100 longer test sequences of 20 frames are generated from this dataset to test the trained networks for a longer prediction horizon. The PredRNN++ [62] architecture, as is discussed in section 3.1, is used for our experiments. The network consists of 4 blocks of causal ConvLSTMs with 12 layers each, with a GHU between the first and second ConvLSTM layers. A 5x5 convolutional kernel size is selected for the ConvLSTM units. The network is trained with each of the losses on 16 mini-batches of 16 samples of the training set per epoch, for 200 epochs. The Adam optimizer [33] is used with a learning rate of $1\mathrm{e}-3$ and an exponential learning rate scheduler that updates the loss by a factor of 0.977 after every epoch. The network is trained to predict five future frames, given five past frames. We evaluate the trained network on the test set for a prediction horizon of 5 and 15 future frames to validate the performance for short and longer term predictions.

Evaluation method The results are evaluated qualitatively by comparing every third frame of a predicted OGM sequence to the ground truth, and quantitatively by comparing the Mean Squared Error (MSE), Image Similarity (IS), Average Precision (AP), and Accuracy metrics, as discussed in section 3.4.

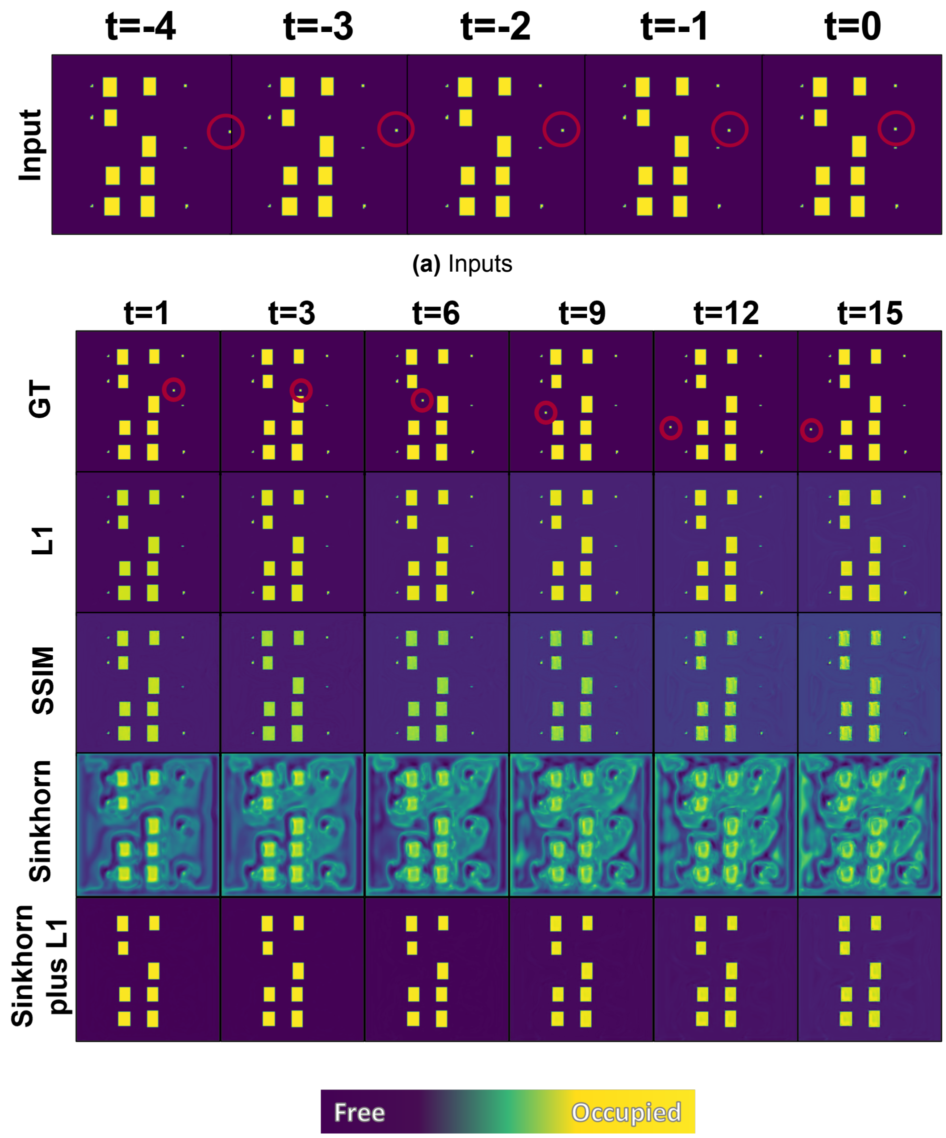
Results Figure 4.7 shows the qualitative results of the toy synthetic OGM prediction network. Table 4.1 summarizes the quantitative results.

Table 4.1: The quantitative comparison on a test sequence from the synthetic dataset for PredRNN++ trained with the L1, SSIM, and Sinkhorn loss functions. The performance is measured using the MSE and IS metrics (lower is better) and AP and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. The L1 loss excels on the majority of the metrics, except the IS on which the SSIM loss performs best.

Loss Function	**Hyperparameters**	**MSE (x10^{-2}) ↓**	**IS ↓**	**AP ↑**	**Accuracy ↑**
T=5					
L1	-	**0.51 ±0.53**	11.367 ±16.104	**0.991 ±0.005**	**0.990 ±0.012**
SSIM	N=M=11	1.07 ±0.90	**1.443 ±0.392**	0.989 ±0.014	0.975 ±0.018
Sinkhorn	ϵ=0.01	7.16 ±2.52	3.266 ±1.674	0.834 ±0.057	0.959 ±0.014
Sinkhorn + L1	ϵ=0.01	1.16 ±0.46	2.807 ±1.422	0.970 ±0.009	0.978 ±0.012
T=15					
L1	-	**0.35 ±0.40**	31.021 ±50.514	**0.988 ±0.005**	**0.995 ±0.009**
SSIM	N=M=11	2.12 ±1.47	**2.150 ±0.972**	0.931 ±0.096	0.963 ±0.019
Sinkhorn	ϵ=0.01	29.51 ±33.10	5.132 ±2.434	0.796 ±0.097	0.956 ±0.014
Sinkhorn + L1	ϵ=0.01	4.77 ±8.28	4.136 ±3.137	0.863 ±0.180	0.915 ±0.147

Discussion Both qualitatively and quantitatively, the L1 loss network performs best except for the IS metric. The L1 loss network's qualitative results show that they resemble the ground truths the most in both the intensity value of the grid cells and the shape of the objects, compared to the SSIM, Sinkhorn and Sinkhorn-L1 networks. Visually, it can be seen that all predictions become more uncertain (shown by the 'greener'/lighter colors) over time and that the blur increases for the SSIM and Sinkhorn loss results. For the Sinkhorn-L1 result, the network shows the least uncertainty over time. The L1 loss result improves for the MSE and the Accuracy for $T = 15$ compared to $T = 5$, whereas the SSIM, Sinkhorn and Sinkhorn-L1 results deteriorate over time. As uncertainty increases over time, deterioration of the results is expected. Why the L1 result improves over time might be because the network has more outliers in the short term predictions and because of the blurring over time, these outliers smooth out for longer-term predictions. The SSIM performs best on the IS score, and the Sinkhorn and Sinkhorn-L1 composite losses also perform better than the L1 loss on the IS score. This could be explained by the distance capturing properties of the inter-pixel loss functions (investigated in the first toy experiment in section 4.1.1). The network could be optimized to maintain inter-pixel relations better when trained with the inter-pixel losses compared to the per-pixel L1 loss. Since the IS also evaluates the OGMs by comparing inter-pixel relations, it can explain why using the inter-pixel losses results in a better performance on the IS score.
The Sinkhorn predictions show well-defined objects as their shape and intensities resemble the ground truths. However, where the ground truth shows free space, uncertain areas are predicted by the Sinkhorn network, especially in between objects. Although the Sinkhorn predictions generally become blurrier and more uncertain over time, at $T = 6$, the prediction shows less uncertainty in the free areas. If this is a trend for all the evaluations, this could explain the the Sinkhorn network's decreased IS at $T = 15$.

Notably, each loss function results in predictions in which the dynamic pedestrian is removed. Based on the sliding square toy experiment of section 4.1.1, this behavior can be expected for the L1 and SSIM losses. The sliding square experiment shows that when the sliding square overlaps an occupied area in the predictions - effectively removing the sliding square as fewer grid cells are now occupied compared to the ground truth - the L1 and SSIM loss values decreased. This shows that removing objects in the predictions is rewarded and can lead the network to a sub-optimal solution. The only situation that would



(a) Inputs

(b) Outputs

Figure 4.7: The qualitative results on a test sequence from the synthetic dataset for PredRNN++ trained with the L1, SSIM, Sinkhorn, and Sinkhorn plus L1 loss functions. The results are shown for time horizon $T = 15$ with an interval of 3 frames. (a) The five input frames to the network, showing eight static vehicles (yellow rectangles), and six static pedestrians (small yellow dots), three on either side of the vehicles. Encircled with red, the dynamic pedestrian is shown on the right, traversing the scene towards the left. (b) Ground Truth with dynamic pedestrian encircled with red (GT) and predictions for the loss functions for selected timesteps. None of the predictions contain the dynamic pedestrian and only predict the static objects, where the Sinkhorn-L1 composite loss also does not predict the static pedestrians. The Sinkhorn loss predictions show uncertainty fluctuations around the predicted objects.

be more rewarded by the L1 and SSIM losses than removing the pedestrian would be if the pedestrian were predicted on a location where it overlaps (partially or exactly) with the ground truth pedestrian. However, due to the uncertainty of the pedestrian's location and its small size, it is deemed likely that the network would predict the pedestrian in a non-overlapping region, resulting in its complete removal. For the Sinkhorn and Sinkhorn-L1 losses, it would be expected that the pedestrian is maintained in the predictions because the Sinkhorn loss is more likely to penalize the removal of objects (as discussed in the sliding square experiment). Perhaps by adding the L1 loss to the Sinkhorn loss, the L1 loss's property of rewarding the removal of the pedestrian was more dominant than the Sinkhorn's penalizing effect. This would cause the network also to remove the pedestrian in the Sinkhorn-L1 predictions. For the Sinkhorn loss alone, the size of the pedestrian might have been too small, compared to the other flaws in the results, to influence the loss significantly enough to optimize the network to predict it. Furthermore, perhaps the speed that the dynamic pedestrian walked was too high for the network to learn its patterns (e.g. crossing the scene). This speed might also have caused a lot of uncertainty since it allowed a large area of possible locations for the pedestrian to be in the following frames, causing the network to disregard it entirely for all loss functions.

Conclusion Based on the qualitative results and most of the metrics, the network trained by the L1 loss shows superior performance on the per-pixel metrics for OGM prediction for both evaluated prediction horizons compared to the inter-pixel losses. For the inter-pixel IS metric; however, the SSIM loss performs best, and the inter-pixel losses outperform the L1 loss for both prediction horizons. Based on these metrics, the L1 loss performs better on a per-pixel level, while the SSIM performs better on an inter-pixel level. The Sinkhorn and Sinkhorn-L1 losses show no superior performance for any metric compared to the other losses, though combining the Sinkhorn with the L1 loss shows a significant increase in performance compared to using the Sinkhorn loss only.

4.2. Occupancy grid map prediction using different loss functions

The two main experiments of this thesis are described in this section and the next one. For each experiment, first, the experiment setup is described, followed by the results. This section describes the first main experiment in which the PredRNN++ [62] network is trained to perform the sequence-to-sequence OGM prediction task using each of the L1, SSIM [64], Sinkhorn [13], and Sinkhorn-L1 composite loss functions. A hyperparameter search study is performed to investigate the effect of the window size when using the SSIM loss and the blur value when using the Sinkhorn and Sinkhorn-L1 composite loss. This hyperparameter search study can be found in Appendix B.

Experimental setup

This experiment evaluates the effects on the test-time performance of training the PredRNN++ OGM prediction network [62] with each of the L1, SSIM, and Sinkhorn losses. Because the Sinkhorn loss is evaluated on normalized data, it is hypothesized that the network trained on the Sinkhorn loss will not learn the right scaling of the OGMs. Therefore, an additional loss that combines the Sinkhorn and L1 loss is used to compensate for the expected effects of normalizing the data. This composite loss is the sum of the Sinkhorn and L1 loss and showed to improve the IS metric in the fourth toy experiment in section 4.1.4. The OGMs are generated from the Waymo Open Perception dataset [60]. This dataset is discussed more in-depth in section 2.1. The code of [61] is used to generate the OGMs from the LiDAR data obtained from the Waymo Open Perception dataset. The object class labels were projected into the OGM to generate the ground truth semantic maps that are used for the semantic OGM prediction in section 4.3. A dataset is generated consisting of 10000 training, 473 validation, and 2972 testing non-overlapping sequences of 10 frames. Additionally, 1405 longer test sequences of 20 frames are generated from this dataset to test the trained networks for a longer prediction horizon. The PredRNN++ [62] architecture, as is discussed in section 3.1, is used for our experiments. The network consists of 4 blocks of causal ConvLSTMs with 12 layers each, with a GHU between the first and second ConvLSTM layers. A 5x5 convolutional kernel size is selected for the ConvLSTM units. The network is trained with each of the losses on 32 mini-batches of 16 samples of the training set per epoch for 200 epochs. The loss hyperparameters are set to a window size of 9x9 for the SSIM loss, a blur of 0.01 for the Sinkhorn loss, and a blur of 1 for the Sinkhorn-L1 composite loss. The Adam optimizer [33] is used with a learning rate of $1\text{e}{-3}$ and an exponential learning rate scheduler that updates the loss by a factor of 0.977 after every epoch. The network is trained to predict five future frames, given five past frames. We evaluate the trained network on the test set for a prediction horizon of 5 and 15 future frames to validate the performance for short and longer-term predictions. The results are evaluated qualitatively by comparing every third frame of a predicted OGM sequence to the ground truth, and quantitatively by comparing the Mean Squared Error (MSE), Image Similarity (IS), Average Precision (AP), and Accuracy metrics, as discussed in section 3.4.

Results

The results for the OGM prediction performance using different loss functions are shown in figure 4.9 qualitatively, and table 4.2 quantitatively. When comparing the quantitative results, the SSIM scores best for all metrics except the Accuracy at $T = 5$ where it is a close second by a thousandth compared to the L1. Compared to the synthetic dataset toy experiment, the L1 loss now performs worse instead of better than the SSIM for most metrics. It can be hypothesized that the SSIM is better at generalizing when more data is used compared to the L1 loss. Another hypothesis is that having more diverse, complex OGM scenes with more dynamic components in the Waymo dataset, compared to the synthetic dataset, requires the network to consider the inter-pixel relations more to generalize well. This would make the inter-pixel SSIM loss more suitable to use compared to the per-pixel L1 loss, which can explain why the SSIM loss performs better than the L1 loss. The Sinkhorn loss scores the worst for all metrics. It was expected that the Sinkhorn loss network would have a scaling problem because it evaluates the normalized outputs and therefore loses information about the range of the grid cell values (between 0 and 1). Figure 4.8 demonstrates this scaling problem. It can be seen that the Sinkhorn network does predict a structure that resembles the structure of the ground truth. However, the range of the grid cell values is too small and not consistent when comparing them with the Sinkhorn predictions in figure 4.9. The Sinkhorn-L1 composite loss performs worse for all metrics compared to the L1 loss, except for the IS and AP metrics at $T = 15$. The result for the IS metric at $T = 15$ demonstrates that the Sinkhorn-L1 composite loss is better at maintaining inter-pixel spatial relations within the OGM for the longer term compared to the L1 loss only, which was also found in the synthetic dataset toy experiment. The standard deviations for each metric are between an order of magnitude one lower or in the same order of magnitude compared to their mean values. Disregarding the Sinkhorn loss, the differences between the mean results are mostly in an order of magnitude that is the same or one below the order of magnitude of the standard deviations.

When looking at the qualitative results, the SSIM shows the least object disappearances and the least blurriness compared to the other results. Besides the SSIM, the L1 and the Sinkhorn-L1 composite losses show that the two vehicles (left in the images) behind the ego-vehicle (center yellow rectangle) are kept in the predictions until $T = 15$. However, the vehicles in front of the ego-vehicle (right in the images) are blurred out or disappear altogether. Then, there is the vehicle next to the ego-vehicle (below the ego-vehicle in the images). In the ground truth, this vehicle overtakes the ego-vehicle, however when looking at the ground truth this vehicle is barely detected during the overtake between $T = 6$ and $T = 9$. In the L1 prediction this vehicle disappears into a fading blur of uncertainty after $T = 3$, while the SSIM, and the Sinkhorn-L1 composite predictions do predict this vehicle up until $T = 15$ and $T = 12$ respectively. Only in the SSIM predictions, this vehicle is predicted in a similar shape to the ground truth. However, the predictions of this vehicle, seem to stay stationary instead of moving, which is not the case in the ground truth. The Sinkhorn loss performs the worst as it seems to only predict the free space in the OGMs, without any other objects or details.

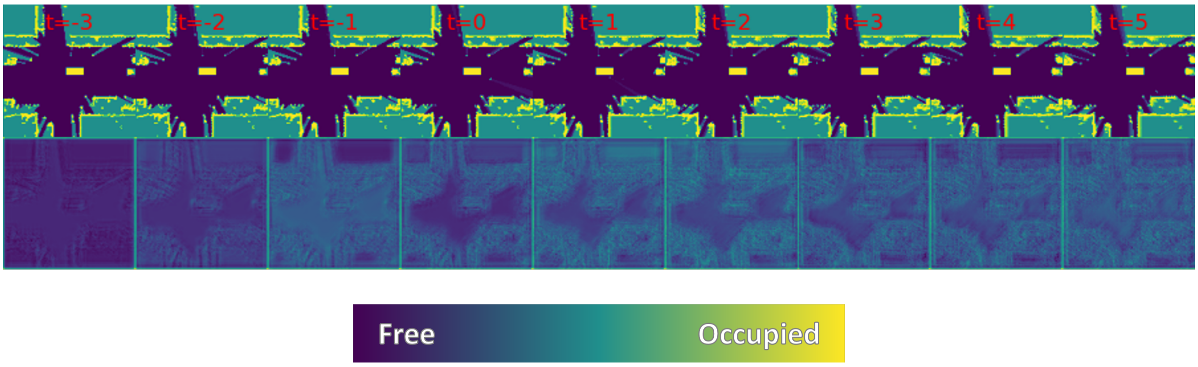
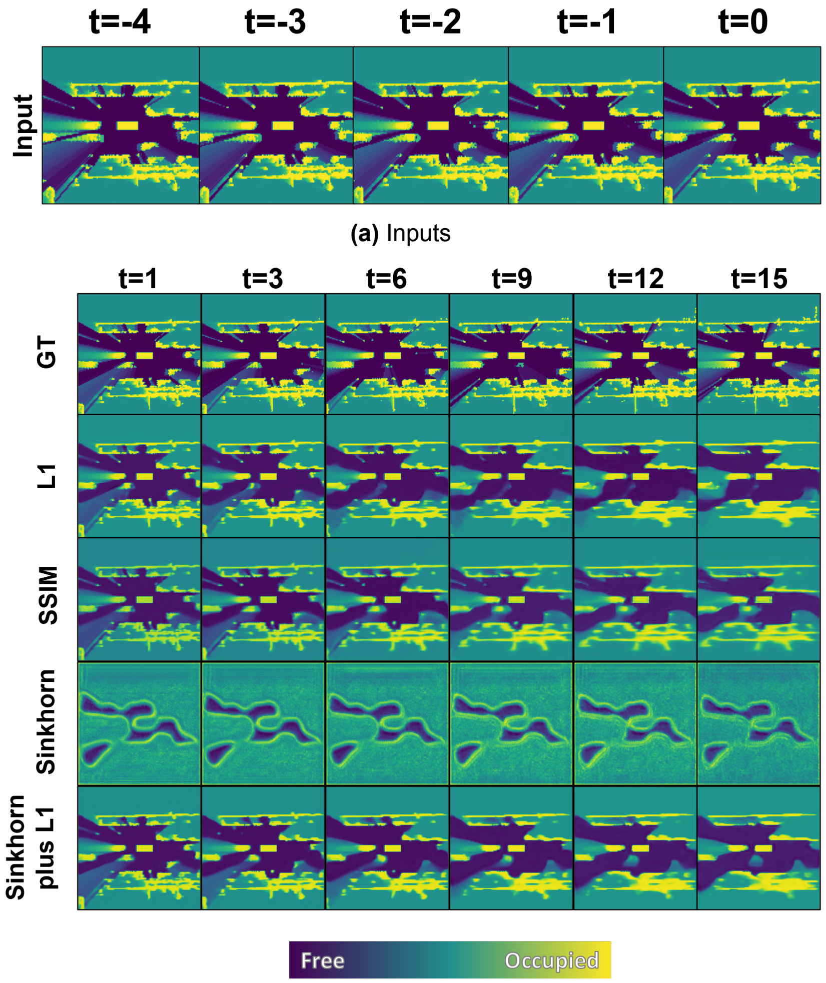


Figure 4.8: An example of a predicted sequence of five OGMs (t=1 to t=5) and the hidden representations of the PredRNN++'s predictions for the past 4 input OGMs (t=-3 to t=0), trained using the Sinkhorn loss. Especially for the hidden representations, the outline of the occupied regions is visible, but at a different occupancy scale range than the ground truth. From t=1, the outline becomes blurrier and noisier.

Table 4.2: The quantitative comparison on the Waymo Perception test set for PredRNN++ trained with various loss functions. Performance is measured using the MSE and IS metrics (lower is better) and AP and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. The SSIM loss excels on the majority of the metrics, except for the Accuracy metric at $T = 5$ on which it scores second with the L1 loss on first place. The Sinkhorn-L1 composite loss scores better than the L1 loss for the IS and AP metrics at $T = 15$. The Sinkhorn loss scores worst on all metrics.

Loss Function	**Hyperparameters**	**MSE (x10^{-2})** ↓	**IS** ↓	**AP** ↑	**Accuracy** ↑
T=5					
L1	-	1.64 ±1.07	1.029 ±0.764	0.948 ±0.050	**0.924 ±0.040**
SSIM	N=M=9	**1.57 ±1.10**	**0.949 ±0.716**	**0.951 ±0.049**	0.923 ±0.046
Sinkhorn	ϵ=0.01	229.79 ±34.59	15.801 ±43.009	0.595 ±0.159	0.398 ±0.172
Sinkhorn + L1	ϵ=1	2.12 ±1.38	1.129 ±0.716	0.943 ±0.053	0.911 ±0.042
T=15					
L1	-	3.38 ±2.44	2.630 ±2.282	0.903 ±0.098	0.862 ±0.088
SSIM	N=M=9	**3.21 ±2.40**	**2.091 ±1.178**	**0.905 ±0.104**	**0.867 ±0.086**
Sinkhorn	ϵ=0.01	226.61 ±33.84	38.781 ±48.661	0.582 ±0.171	0.397 ±0.172
Sinkhorn + L1	ϵ=1	4.70 ±3.41	2.179 ±1.535	0.904 ±0.095	0.853 ±0.081



(a) Inputs

(b) Outputs

Figure 4.9: The qualitative results on a Waymo Open Perception test sequence for PredRNN++ trained with the L1, SSIM, Sinkhorn, and Sinkhorn plus L1 loss functions. Results are shown for $T = 15$. (a) The five input frames to the network, showing three vehicles approaching the ego-vehicle (central yellow rectangle) from behind (left of image), and three vehicles moving away from the ego-vehicle (right). (b) Ground Truth (GT) and predictions for the loss functions for selected timesteps. The per-pixel L1 loss leads to more object disappearances, and blurriness over long time horizons ($T = 15$) compared to the other losses except the Sinkhorn loss. The Sinkhorn loss predictions show mostly uncertainty (green color), where only the free areas (dark blue) are predicted in a similar shape compared to the ground truth's free areas. The SSIM seems to show the least disappeared objects up until $T = 15$ compared to the other losses.

4.3. Multi-task learning of Occupancy Grid Map prediction and semantic segmentation

This experiment investigates the effect of multi-task learning the OGM prediction and the semantic segmentation tasks on the performance of the OGM prediction task. The PredRNN++ [62] network is given a sequence of past evidential OGMs to predict semantic channels for every frame in the sequence of future evidential OGMs. To perform this experiment, the last convolutional layer of the PredRNN++ is modified to output seven channels (one for the occupancy values and six for the semantic channels) instead of one channel. Per evidential OGM, six binary semantic OGM channels are generated from the labeled LiDAR data from the Waymo [60] dataset, which togethers form the ground truth to the network's seven output channels. The experimental setup is described first, followed by the results for the evidential OGM prediction. The results for the predicted semantic segmentation channels are presented in Appendix C.

Experimental setup

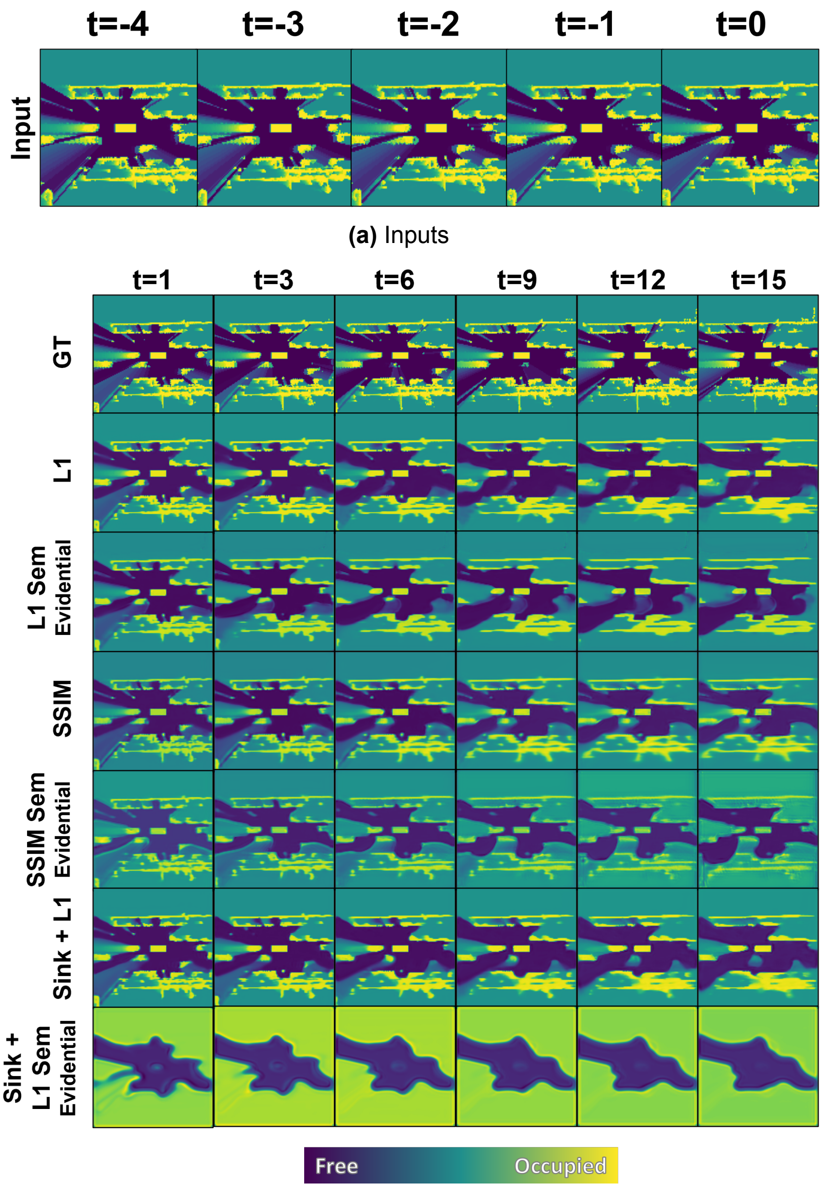
The evidential OGMs with six additional semantic channels are generated from the Waymo Open Perception dataset [60], using an adapted version of [61]'s code to obtain labels for the semantic classes. Six different semantic class labels are extracted per frame and projected onto separate semantic OGM channels. Each grid cell in the semantic channels corresponds to a grid cell in the evidential OGM. Together the semantic OGMs represent per grid cell of the evidential OGM what the probability is that that grid cell belongs to one of the following six semantic classes: free, uncertain, static occupied, vehicles, pedestrians, or cyclists. The dataset with the same evidential OGM sequences from the loss experiments in section 4.2 are used, but with the additional six semantic channels for the training and validation set. The Waymo Open dataset's test set has no class labels, so for the test set, only the evidential OGMs are used to evaluate the predictions. The dataset consists of 10000 training, 473 validation, and 2972 testing non-overlapping sequences of 10 frames, with an additional 1405 sequences of 20 frames for evaluation of longer prediction horizons. The same PredRNN++ architecture and training settings are used as in the loss experiments, except that the PredRNN++ is adjusted to output seven channels (one evidential and six semantic channels) instead of one evidential channel. Given five past single-channel evidential frames, the network is trained to predict five future frames consisting of the seven channels. Three networks are trained, using the L1 loss, SSIM loss with a window size of 9x9, and the Sinkhorn loss with a blur of 1, respectively. Then, the trained network is evaluated on the test set for a prediction horizon of 5 and 15 future frames. The results are evaluated qualitatively on the evidential channels of the 7-channel outputs, by comparing every third frame of a predicted evidential OGM sequence to the ground truth. Quantitatively, the Mean Squared Error (MSE), Image Similarity (IS), Average Precision (AP), and Accuracy metrics are used for evaluation of the evidential OGM channels, as discussed in section 3.4. The results are compared to the results from the single-task networks trained using the different losses. Furthermore, the semantic segmentation performance is evaluated on the validation dataset. Firstly, because the test set does not contain semantic segmentation ground truths. Secondly, because the validation set is not used to adjust any hyperparameters during training and is thus still deemed independent from the training process. Thirdly, because the semantic segmentation task is not the primary objective of this experiment, but rather the performance on the evidential OGM prediction task when the network is trained on semantic segmentation. These semantic segmentation results are also evaluated qualitatively, using images, and quantitatively, using the above-mentioned metrics per semantic channel in Appendix C.

Results evidential OGM prediction

Figure 4.10 shows the evidential predictions by the PredRNN++ network for each loss trained only for OGM prediction (L1, SSIM, Sink + L1), and trained for multi-task learning (L1 Sem Evidential, SSIM Sem Evidential, Sink + L1 Sem Evidential). The quantitative results are shown in table 4.3. The quantitative results show that for each loss function, the multi-task trained networks generally perform worse than the networks trained for OGM prediction only. There are some exceptions. The L1 Semantic network performs better, by a thousandth than the L1 network for the AP metric at $T = 15$. The SSIM Semantic network performs better than the SSIM for the Accuracy at $T = 15$. The Sinkhorn-L1 Semantic network performs better than the Sinkhorn-L1 network for the IS metric at $T = 5$. The qualitative results show that the predictions from the L1 Semantic network retain objects for a longer time before they disappear or blur compared to the L1 network. The SSIM Semantic predictions show more consistent object shapes but also more uncertainty of all the occupancy states as the prediction horizon increases, compared to the SSIM network. The predictions by the Sinkhorn-L1 Semantic network lack any detail and only show either occupied or free grid cell states. Therefore the Sinkhorn-L1 Semantic network shows no improvement compared to the Sinkhorn-L1 network predictions.

Table 4.3: The quantitative comparison on the Waymo Perception test set for PredRNN++ trained for evidential OGM prediction only (L1, SSIM, and Sinkhorn-L1) and multi-task trained for evidential and semantic segmentation prediction (L1 Semantic, SSIM Semantic, and Sinkhorn-L1 Semantic). Performance is measured using the MSE and IS metrics (lower is better) and AP and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. The SSIM performs best on the MSE, IS, and AP for both prediction horizons. The L1 performs best for the Accuracy at $T = 5$ and the SSIM Semantic for the Accuracy at $T = 15$. The L1 Semantic performs worse than the L1 on all metrics except for the AP at $T = 15$. The SSIM Semantic performs worse than the SSIM on all metrics except for the Accuracy at $T = 15$. The Sinkhorn-L1 Semantic performs worse than the Sinkhorn-L1 on all metrics except for the IS at $T = 5$.

Loss Function	**Hyperparameters**	**MSE (x10^{-2}) ↓**	**IS ↓**	**AP ↑**	**Accuracy ↑**
T=5					
L1	-	1.64 ±1.07	1.029 ±0.764	0.948 ±0.050	**0.924 ±0.040**
L1 Semantic	-	1.99 ±1.16	1.312 ±0.936	0.945 ±0.052	0.906 ±0.043
SSIM	N=M=9	**1.57 ±1.10**	**0.949 ±0.716**	**0.951 ±0.049**	0.923 ±0.046
SSIM Semantic	N=M=9	2.46 ±1.41	1.449 ±0.934	0.926 ±0.068	0.882 ±0.056
Sinkhorn + L1	ϵ=1	2.12 ±1.38	1.129 ±0.716	0.943 ±0.053	0.911 ±0.042
Sinkhorn + L1 Semantic	ϵ=1	6.95 ±1.81	1.037 ±0.395	0.870 ±0.120	0.790 ±0.056
T=15					
L1	-	3.38 ±2.44	2.630 ±2.282	0.903 ±0.098	0.862 ±0.088
L1 Semantic	-	3.90 ±2.59	2.988 ±2.461	0.904 ±0.099	0.852 ±0.080
SSIM	N=M=9	**3.21 ±2.40**	**2.091 ±1.178**	**0.905 ±0.104**	0.867 ±0.086
SSIM Semantic	N=M=9	5.98 ±4.22	4.074 ±3.058	0.868 ±0.128	**0.882 ±0.216**
Sinkhorn + L1	ϵ=1	4.70 ±3.41	2.179 ±1.535	0.904 ±0.095	0.853 ±0.081
Sinkhorn + L1 Semantic	ϵ=1	10.04 ±3.96	2.495 ±2.316	0.835 ±0.144	0.662 ±0.173



(a) Inputs

(b) Outputs

Figure 4.10: Qualitative results on a Waymo Open Perception test sequence for PredRNN++ trained using different losses, with and without multi-task performing the semantic segmentation task besides the OGM prediction task. Results are shown for $T = 15$. (a) The five input frames to the network, showing three vehicles approaching the ego-vehicle (central yellow rectangle) from behind (left of image), and three vehicles moving away from the ego-vehicle (right). (b) Ground Truth (GT), the predictions for the losses without performing the semantic segmentation task (L1, SSIM, Sink + L1) and the evidential predictions with performing the semantic segmentation task (L1 Sem Evidential, SSIM Sem Evidential, Sink + L1 Sem Evidential). When comparing the L1 and L1 Sem Evidential predictions, one can see that the latter retains the vehicles for a longer prediction horizon or blurs the vehicles where they disappear in the former. When comparing the SSIM and SSIM Sem Evidential predictions, the latter's free and occupied regions seem less certain (more towards the green color), but the objects have a constant shape and are not blurred out over time. The Sinkhorn + L1 Sem Evidential predictions show that the composite loss does not perform well for multi-task learning. Only free and occupied regions are predicted without any defined objects or uncertain regions.

5 Discussion

This thesis posits that blurring and object disappearances in the state-of-the-art OGM prediction methods is a major issue that affects an AV's ability to anticipate road user behavior for long prediction horizons and thus affects the traffic safety. One could pose that blurring OGMs might be desirable when predicting, for instance, the future road layout because there can be sudden curves or crossings. Blurring allows the AV to anticipate a change in road layout better than predicting a continuous straight road. However, objects are expected to stay rigid and to move smoothly, as stated by [66]. In this thesis, it is therefore expected that blurring or removing objects from the predictions does not benefit the AV in anticipating their behavior and ensuring their safety. Therefore, two methods to improve the OGM sequence-to-sequence predictions by reducing the blurring and the disappearance of objects are investigated in this thesis.

The first proposed solution is to investigate the effects of the SSIM [64], Sinkhorn and Sinkhorn-L1 composite [13] inter-pixel loss functions, compared to the L1 per-pixel loss, on training the PredRNN++ [62] network to perform OGM predictions. The expectation is that inter-pixel losses can train the network to maintain better inter-pixel spatial relations in the OGM predictions. This would mean that objects in OGMs, which are clusters of occupied grid cells that have a specific spatial relation, are better maintained which would reduce their disappearance and blurring in the predictions. The SSIM and Sinkhorn-L1 inter-pixel losses both show these improvements quantitatively and qualitatively compared to the L1 loss. The second proposed solution is to investigate if the principle of multi-task learning can be applied to improve the OGM predictions. The PredRNN++ network is trained to perform the additional semantic segmentation task besides the OGM predictions. It is expected that the network will learn latent object representations, which it can use to maintain the object instances in the predictions better. The results in this thesis demonstrate that this method does not improve the predictions quantitatively. Qualitatively, there are some improvements when using the L1 and SSIM losses to train the multi-task network. This chapter provides a thorough discussion of each experiment and the results from this thesis with regard to answering the proposed research questions.

Does the use of an inter-pixel loss function, such as the Structural Similarity Index Measure loss or the Sinkhorn loss, to train a sequence-to-sequence Occupancy Grid Map prediction network improve its performance, and reduce blurriness and objects disappearance compared to using a per-pixel loss?

Compared to the L1 per-pixel loss, the SSIM loss results in an improvement of the predictions both qualitatively by generating less blur and fewer object disappearances, and quantitatively according to the reported metrics. Using the SSIM loss, with a window size of 9x9 pixels, results in a better or equal performance for all metrics (table 4.2) and qualitatively (figure 4.9) compared to the L1 loss. That the SSIM performs the best compared to the other losses on the IS metric could be explained by the SSIM's property to evaluate perceptual similarities, which the IS metric also evaluates. It is hypothesized that the SSIM optimizes the network to generate perceptually similar OGMs and thus also optimizes the network to perform well on the IS metric. Lange, Itkina, and Kochenderfer [35] state that when a network is optimized for the per-pixel MSE metric, which the L1 is more likely to do due to its per-pixel property, uncertainty is often expressed as a blur to prevent large errors. On the other hand, when a network is optimized for the IS metric, which the SSIM is more likely to do because of its inter-pixel and perceptual similarity properties, object shapes are maintained for longer-term predictions and blurring would be limited. So, training a network with the L1 loss is expected to increase the MSE and decrease the IS metrics, where the opposite is expected for the SSIM loss. However, unexpectedly according to the results from the synthetic dataset toy experiment in section 4.1.4, for the SSIM loss with the 9x9 pixel window size, both the IS and the MSE values are better than the results for the L1 loss. This suggests that the L1 loss, despite its per-pixel property, is not always the best optimizer for the MSE metric. Perhaps the L1 loss network's generated blur is so large that it affects the MSE more than any misplaced object generated by the SSIM network. Since the AP and Accuracy metrics are used for object detection evaluation, the SSIM's ability to maintain objects for longer-term predictions with less blur is expected to benefit these metrics. This can explain the improved or equally good performance of the SSIM loss compared to the L1 loss for the AP and Accuracy metrics. The visual qualitative improvement compared to the L1 loss is in line with [64]'s philosophy behind the design for the SSIM metric, which states that the SSIM evaluates the similarity between images (OGMs) based on how human perception functions. The SSIM network is therefore optimized to generate more similar OGMs according to human perceptual evaluation, which is demonstrated in the figures. This effect is also displayed in the noise-image-to-Gaussian toy experiment in section 4.1.3.

Then, the results for the Sinkhorn loss (table 4.2) show that the Sinkhorn loss alone is not a suitable loss to train the OGM network. As mentioned before and shown in figure 4.8, because the Sinkhorn loss is performed on normalized outputs, the network does not learn at what scale it should generate its outputs. This scaling problem is solved in the experiments by combining the Sinkhorn loss with the L1 loss. This Sinkhorn-L1 loss combination is formed by taking their sum. Quantitatively it only outperforms the L1 loss for the IS and AP metrics at $T = 15$. This is unexpected when also considering the results of the sliding square toy experiment in section 4.1.1. This experiment shows that, unlike the L1 loss, the Sinkhorn loss can capture the distance errors for objects in the scene and continuously increases its loss value with the distance error while it also penalizes object disappearance. In combination with the L1 loss, the expectation was that the Sinkhorn part optimizes the network to get the distance error to zero, while the L1 part optimizes the outputs to be in the correct scaling. However, the pdf to cdf toy experiment in section 4.1.2, and the noise-to-Gaussian conversion toy experiment in section 4.1.3, showed that the Sinkhorn loss converges the networks to generate outputs with larger variances or scatterings compared to the L1 and SSIM losses. The synthetic dataset toy experiment in section 4.1.4 also showed an inferior performance by the Sinkhorn and Sinkhorn-L1 losses. Perhaps the blurring of the Sinkhorn loss has a limiting effect on the accuracy that the OGM prediction network can achieve. Also, the Sinkhorn and L1 losses are summed with equal weights in the Sinkhorn-L1 composite loss. It could be that the Sinkhorn loss was not complemented by the L1 loss, which accounts for the scaling information, but instead counteracted the inter-pixel properties of the Sinkhorn loss. Perhaps if the Sinkhorn and L1 parts of the Sinkhorn-L1 composite loss were differently weighted, the inter-pixel properties of the Sinkhorn would improve the results for more metrics. However, the improved results of the IS and AP metrics for the long-term predictions by the Sinkhorn-L1 loss, compared to the L1 loss, could mean that the Sinkhorn-L1 loss only performs better for longer-term predictions. Perhaps the inter-pixel properties of the Sinkhorn-L1 loss trains the network to maintain objects for longer prediction horizons, while the L1 loss predictions blur them out, which would affect the inter-pixel IS and AP scores. The qualitative results of the Sinkhorn-L1 loss also show that objects are maintained for a longer prediction horizon compared to the L1 loss.

Does the performance of an Occupancy Grid Map prediction network improve by means of multi-task learning if it also learns to perform semantic segmentation on the predictions?

Besides investigating the effects of loss functions, this thesis also investigates whether multi-task learning, i.e. learning the semantic segmentation task to improve the OGM predictions, is possible. By training the OGM prediction network to generate both future OGM predictions and their respective semantic class channels, it is hypothesized that the predictions improve both qualitatively and quantitatively compared to the network that does not perform the additional semantic segmentation task. However, the results in figure 4.10 and table 4.3 show the opposite for all losses. Dequaire et al. [15] has demonstrated in their research that their network's performance on the semantic classification task improved when they trained the network on a tracking task first. The network learned object representations when learning the tracking task, which it could use to perform better at the classification task. This thesis shows that multi-task learning does not improve the OGM prediction network's performance by learning the semantic segmentation task for most metrics at both prediction horizons of $T = 5$ and $T = 15$. It was hypothesized that the network would learn hidden object representations from the semantic segmentation task, which it would use to generate predictions, with reduced blurring, that maintain objects for longer prediction horizons compared to a single-task network that is trained to perform the OGM prediction task. The only result that demonstrates this hypothesis, both quantitatively and qualitatively is shown by the multi-task SSIM Semantic network at $T = 15$ where it outperforms all other networks on the Accuracy metric, and it shows less blur and more maintained objects compared to the single-task SSIM network's results. The L1 Semantic network also shows better qualitative results, however, this is not shown by any of the metrics. On the other hand, the Sinkhorn-L1 Semantic network performs better on the IS score at $T = 5$, which is however not visible in the qualitative results. Based on the other metrics at both prediction horizons, it cannot be confirmed that multi-task learning the semantic segmentation task and the OGM prediction task will improve the latter. There are several hypotheses as to why multi-task learning did not improve the predictions.

First, as described in the Preliminaries in section 1.4.4, the expectation of multi-task learning is that learning a source task, i.e. semantic segmentation in this thesis, has a regularizing effect on learning the target task, i.e. OGM prediction, by adjusting the model's parameters to be useful for both tasks, preventing the parameters from overfitting on the target task. However, this expectation assumes that there is ample data available to learn the source task sufficiently to regularize the network and improve the target task. In this thesis' experiments, for some classes there were only few data available. The Waymo Open perception dataset [60] does not contain many cyclists and pedestrians. For the 10000 training and 473 validation data samples used in the experiments, where each sample consists of 10 timesteps (frames), table 5.1 shows how the six investigated semantic classes are distributed over the samples, frames, and individual grid cells. In the training set, only about 45% and 6% of the samples and frames contain pedestrians and cyclists, respectively. For the individual grid cells that contain these classes, it can be seen that the Free space and Unknown classes represent the majority of the grid cells with the respective % and % occurrence. The number of grid cells is much lower for the other classes, which is expected to limit the networks' ability to learn to predict those classes. This is also shown in the semantic segmentation results in Appendix C. Tables C.1, C.2, and C.3 show that the AP scores for vehicles and static objects are all lower than those for the Free space and Unknown classes. The scores for pedestrians and cyclists are even lower and close to zero. The network does not learn to predict these semantic classes well. In turn, the network might not have accurately learned the hidden object representations. Instead of improving, multi-task learning the semantic segmentation task could have deteriorated the OGM task because of the lack of data for each class.

Second, the single-task network, trained only for OGM prediction, and the multi-task network have an equal number of trainable parameters, except for the last convolutional layer in the multi-task network, which generates the six additional semantic channels compared to the single-task network. Perhaps predicting all the semantic channels by the multi-task network requires more predictive capacity than can be achieved by the current number of trainable parameters to reach the same or better performance for the longer-term predictions compared to the single-task network. Alternatively, having only one convolutional layer between the last hidden layer and output could have been too little for the seven output channels to differentiate from each other and generate the desired outputs.

Table 5.1: The total counts and percentages for each semantic class occurrence for each sample, each frame, and each grid cell in the training split and validation split. The total number of samples in the splits are 10000 and 473 with each 10 frames (timesteps), containing 128x128 grid cells, per sample, respectively.

Channel	Training Set						Validation Set					
	#Samples		#Frames		#Grid Cells		#Samples		#Frames		#Grid Cells	
Free Space	10K	100%	100K	100%	581.94M	35.52%	0.47K	100%	4.73K	100%	30.60M	39.5%
Vehicles	10K	100%	100K	100%	53.20M	3.25%	0.47K	100%	4.73K	100%	2.28M	2.9%
Pedestrians	4.78K	47.8%	45.47K	45.5%	3.47M	0.21%	0.22K	46.1%	2.08K	43.9%	86.55K	0.1%
Cyclists	0.69K	6.9%	6.13K	6.1%	0.13M	0.01%	0.03K	6.1%	0.24K	5.1%	5.33K	0.01%
Static Objects	10K	100%	100K	100%	133.70M	8.16%	0.47K	100%	4.73K	100%	5.83M	7.5%
Unknown	10K	100%	100K	100%	866.07M	52.86%	0.47K	100%	4.73K	100%	38.69M	49.9%

Third, using the same unweighted losses to evaluate both the OGM and the semantic channels could cause the network to optimize for the semantic channels more than for the OGM prediction. Especially because of the lack of data for some classes, the network might not (learn to) perform well on those classes, causing the loss to be higher for those channels compared to the channels for which more data is available. The network might then overfit the hidden layers on the semantic classes represented the least in the data. This, in turn, could affect the quality of the OGM predictions. Perhaps this has caused the Sinkhorn-L1 multi-task network to predict almost binary OGMs, lacking most objects (figure 4.3). The sliding square toy experiment (in section 4.1.1) shows that the Sinkhorn loss increases more if the number of occupied grid cells in the ground truth do not match the number of occupied grid cells in the prediction (when the sliding square overlaps the stationary square in the experiment's second scene). It can be hypothesized that this caused high losses for the free space and unknown semantic channels since the majority of the OGMs consist of those classes. This could have caused the network to overfit on predicting those channels, which affected the OGMs to only differentiate between two states.

Although the quantitative results of the multi-task networks show that the OGM predictions do not improve compared to using the single-task networks, some improvement can be seen in the qualitative results. Figures C.1b and C.2b in Appendix C show that L1 and SSIM loss networks predict the vehicle class accurately when they are close to the ego-vehicle. Notably, the SSIM network only predicts outlines of the vehicles. When observing the qualitative results of the OGM predictions in figure 4.3, it can be seen that both the L1 Semantic and SSIM Semantic predictions maintain vehicles for a longer prediction horizon compared to the respective single-task networks. This could be the effect of multi-task learning, where accurately predicting the semantic vehicle class also improves the occupancy predictions of that class.

Concerning all the results in this thesis, there are some limitations to using the Waymo dataset. As also described in [35], the Waymo dataset contains many scenes located on highways. These scenes contain a lot of straight roads with mostly cars and a lack of pedestrians and cyclists. The trained networks will be biased towards these highway traffic scenes when using this dataset. Lange, Itkina, and Kochenderfer [35]'s research shows that their proposed network architecture performs much better using the Waymo Perception dataset, compared to the smaller, more diverse KITTI [22] dataset. Although, the smaller size of KITTI plays a role in the worse performance, the KITTI dataset also contains more urban and suburban traffic scenes, making the dataset more diverse and less generalizable than the Waymo dataset. Performing the experiments in this thesis on another large, diverse dataset should be done to investigate if this research's results also apply to other datasets.

The standard deviations for the quantitative results are mostly between the same order or one order of magnitude lower than the mean results. This high variability of the networks' predictions can mean that on a different test set or using a different dataset the result might lead to different conclusions. Therefore, the results of this thesis would become more convincing if these same experiments were repeated for different datasets and different network architectures.

6 Conclusion

This thesis investigates two methods to solve the problem of blurring and object disappearances in the long-term predictions generated by OGM prediction methods. The effects of using inter-pixel loss functions and multi-task learning the additional semantic segmentation task on the quality of the OGM predictions are researched. This chapter answers the research question proposed in section 1.2.

To answer the first research question, this thesis investigated the effects of several inter-pixel loss functions on the performance of OGM prediction by the PredRNN++ network and compared them to the per-pixel L1 loss baseline performance. The main conclusion from the experiments is that using an inter-pixel loss function can improve the network's prediction performance for the MSE, IS, AP, and Accuracy metrics and that using an inter-pixel loss reduces blurriness and disappearance of objects in long term predictions compared to the L1 loss baseline. Using the SSIM loss on the PredRNN++ to perform OGM predictions improves the predictions for the MSE, IS, and AP metrics, for both prediction horizons of $T = 5$ and $T = 15$, and improves the Accuracy metric for $T = 15$, when using a window size of 9x9 pixels, compared to the L1 baseline. Furthermore, as hypothesized in this thesis's introduction, the SSIM loss reduces blurriness and object disappearances for longer term predictions with respect to the L1 loss baseline. Using the isolated Sinkhorn loss to train the PredRNN++ for OGM prediction does not result in good predictions. The predictions are not scaled right because the Sinkhorn loss evaluates the normalized outputs of the network. Therefore, the network does not learn in what range to generate predictions. The Sinkhorn plus L1 composite loss provides a solution because the L1 loss compensates for the scaling errors. The results show that the Sinkhorn-L1 composite loss outperforms the L1 baseline for the IS and AP metric for a prediction horizon of $T = 15$. The Sinkhorn-L1 composite also shows to reduce blurriness and object disappearance for longer term predictions compared to the L1 loss baseline.

To answer the second research question, based on the experiment on multi-task learning performed in this thesis, the PredRNN++'s performance on OGM prediction does not improve for most of the investigated metrics when it is also trained to perform the semantic segmentation task. Qualitatively, the semantic segmentation networks, trained using the L1 and SSIM losses, show that for longer term predictions it can better retain the shapes of dynamic objects that it has classified correctly compared to the L1 loss baseline.

All in all, the results of this research show that the use of inter-pixel losses can improve the quality of OGM prediction networks, while performing multi-task learning does not result in improved OGM prediction networks. Reducing blurriness and the disappearance of objects for long-term predictions has been a challenge in object agnostic OGM prediction methods and this thesis provides part of a solution to solve this problem. Hereby this research contributed to improving the safety of AVs. This thesis lays a foundation for future research in motion prediction. Moreover, the video prediction domain also struggles with blurriness in long term predictions and since sequence-to-sequence prediction of OGMs is similar to video prediction, this thesis also provides a contribution to the video prediction domain.

7 Future Work

This chapter provides some recommendations for future research that can be performed to develop motion prediction methods for AVs. First, some recommendations within the scope of this thesis regarding loss functions and multi-task learning are provided. Then, some recommendations regarding the domain of object-agnostic motion prediction are discussed.

7.1. Future work on inter-pixel loss functions

The results from this thesis inspire the search for more possibilities to use inter-pixel loss functions for OGM prediction. In the experiments, the Sinkhorn-L1 composite loss is used to benefit from the Sinkhorn's distance-capturing properties while compensating for the scaling errors using the L1 loss. However, the experiments only investigate the unweighted sum of those losses, while there are more possible ways to combine the Sinkhorn and L1 losses, e.g. by adding a weight to the L1 loss. The scaling problem can also be compensated without using the L1 loss. Research could be done on more Sinkhorn composites with, for instance, the SSIM loss, the L2 loss, or the Cross-Entropy loss. Furthermore, the use of a perceptual loss network as proposed by [29] could be investigated. Johnson, Alahi, and Fei-Fei [29] proposes to use a CNN pre-trained on image classification to act as a loss function. The feature representations of the predictions and the ground truths that the loss network outputs are compared. Their research found that this method works similar to using an inter-pixel loss function. Also, this thesis's experiments should be expanded by testing the hypotheses on multiple datasets and multiple sequence-to-sequence prediction networks. Not only within the OGM prediction domain but also in the closely related video prediction domain. This way, the effects of using inter-pixel loss functions found in this thesis can be reinforced and verified on a larger scale for multiple applications.

7.2. Future work on multi-task learning

It was shown in the discussion that the Waymo [60] dataset does not have many representations of the pedestrian and cyclist classes. This could have resulted in the deteriorated performances of the multi-task networks compared to the single-task network. The multi-task learning experiments can be investigated on different, more diverse datasets with a larger number of labeled objects, such as the nuScenes [7], ECP2.5D [5], or Argoverse [8] datasets described in section 2.1 datasets. Perhaps using these datasets for multi-task learning will improve the OGM predictions. However, most large-scale real-world datasets suffer from class imbalances. This means using a different dataset might not solve this scarcity problem of certain object classes [12] and would thus not improve the multi-task network performance. Cui et al. [12] provide a theoretical framework to use a class-balanced loss function which has shown to significantly improve the performance of visual recognition tasks. In future research, this theoretical framework could be extended to the L1, SSIM, and Sinkhorn-L1 loss functions to train the multi-task network for the semantic segmentation and OGM prediction tasks. If using a class-balanced loss improves the semantic segmentation task, it might also improve the OGM prediction task.

Furthermore, it is hypothesized that the multi-task network performs worse than the single-task network because of limited trainable parameters in the network to perform both the OGM prediction and the semantic segmentation tasks accurately. Future research can investigate if the performance for longer-term predictions improves if larger network architectures are used, or if additional semantic segmentation heads are added to the PredRNN++ network.

Also, in this thesis, each semantic channel is evaluated separately to their respective ground truths. In future research, the inter-pixel losses could be extended also to measure any relations between the semantic channels, or between a semantic channel and the corresponding OGM. For example, the SSIM loss could be extended by using 3D boxes that compute the loss over all predicted channels simultaneously instead of using 2D windows that can only compute the loss of one channel at a time. Also, an ablation study can be done to investigate what semantic class has the most influence on improving the OGM prediction task.

Then, the goal of multi-task learning for OGM prediction is to train the network on a certain task that allows it to learn object representations so it will not blur or remove the objects for longer-term predictions. These representations can also be learned through other means, such as attention as proposed by Lange, Itkina, and Kochenderfer [35]'s Attention Augmented ConvLSTM (AAConvLSTM) network. Future work could investigate whether the attention architecture can also be used to enhance the effects of the semantic segmentation and OGM prediction multi-task learning.

7.3. Future work on object-agnostic motion prediction

In 2D OGM prediction, 3D measurements (often LiDAR point cloud data) are reduced to a 2D BEV representation of the environment. Through this process, a lot of spatial data is lost that might be key to generating better object representations of the environment when regarding OGM prediction networks. Degerman, Pernstål, and Alenljung [14], however, proposed a way to generate 3D OGMs. Future research could be done on 3D sequence-to-sequence OGM prediction by generating 3D OGM sequences from LiDAR data. State-of-the-art architectures that can process 3D representations, such as Guo et al. [23]'s ST-3DNet, or Lin, Gan, and Han [38]'s more efficient Temporal Shift Module, can be adjusted to process 3D sequences.

Finally, although ensuring safe behavior by AVs is the main motivation of this thesis and the research found in literature on object-agnostic motion prediction, there does not seem to be a consensus on how to measure the 'safety' of the predictions. When looking a table 2.2 in the Related Work, many different metrics are used to evaluate the performance of the prediction methods. Both per-pixel and inter-pixel metrics are used, some more generally used in deep learning and some more specifically used in the object detection domain. Also, in this thesis, multiple metrics were used to evaluate the OGM predictions, where the inter-pixel IS metric sometimes ranks the network performances differently than the per-pixel MSE metric. In those contradictory cases, it is hard to conclude based on the metrics which network would be better or safer to use in AV applications. None of the metrics, however, look at what the proposed path is the AV takes when it uses the predictions and whether that path is considered safe. Therefore, the final recommendation of this thesis is that research should be done on defining and evaluating safety regarding motion prediction. If a consensus on this can be reached, future research can focus better on optimizing for that definition of safety. This way, the results will be more comparable, making it easier to conclude what prediction method is safer.

References

[1] National Highway Traffic Safety Administration et al. "Critical reasons for crashes investigated in the national motor vehicle crash causation survey". In: *Washington, DC: US Department of Transportation* 2 (2015), pp. 1–2.

[2] Alan Ohnsman. *Cruise Looks To The Skies As It Readies Robotaxi Service*. 2021. URL: https://www.forbes.com/sites/alanohnsman/2021/09/08/cruise-looks-to-the-skies-as-it-readies-robotaxi-service/?sh=7c6a12ef2a5b.

[3] Andreas Birk. "Learning geometric concepts with an evolutionary algorithm". In: *environment* 4 (1996), p. 3.

[4] Andreas Birk and Stefano Carpin. "Merging occupancy grid maps from multiple robots". In: *Proceedings of the IEEE* 94.7 (2006), pp. 1384–1397.

[5] Markus Braun, Sebastian Krebs, and Dariu M Gavrila. "ECP2. 5D-Person localization in traffic scenes". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 1694–1701.

[6] *C2RMF, Galerie de tableaux en très haute définition*. https://c2rmf.fr/imagerie. Accessed: 2022-03-13.

[7] Holger Caesar et al. "nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[8] Ming-Fang Chang et al. "Argoverse: 3d tracking and forecasting with rich maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8748–8757.

[9] Fang-Chieh Chou et al. "Predicting motion of vulnerable road users using high-definition maps and efficient convnets". In: *arXiv preprint arXiv:1906.08469v2* (2020).

[10] Henggang Cui et al. "Multimodal trajectory predictions for autonomous driving using deep convolutional networks". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 2090–2096.

[11] Jin Cui et al. "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles". In: *Ad Hoc Networks* 90 (2019), p. 101823.

[12] Yin Cui et al. "Class-balanced loss based on effective number of samples". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9268–9277.

[13] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

[14] Johan Degerman, Thomas Pernstål, and Klas Alenljung. "3D occupancy grid mapping using statistical radar models". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 902–908.

[15] J. Dequaire et al. "Deep tracking in the wild: End-to-end tracking using recurrent neural networks". In: *The International Journal Of Robotics Research* 37 (2018), pp. 492–512.

[16] Nemanja Djuric et al. "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 2095–2104.

[17] Alberto Elfes. "Using occupancy grids for mobile robot perception and navigation". In: *Computer* 22.6 (1989), pp. 46–57.

[18] J Feydy. "Geometric data analysis, beyond convolutions". PhD thesis. PhD thesis, Université Paris-Saclay, 2020.

[19] Chelsea Finn, Ian Goodfellow, and Sergey Levine. "Unsupervised learning for physical interaction through video prediction". In: *Advances in neural information processing systems* 29 (2016), pp. 64–72.

[20] R'emi Flamary et al. "POT: Python Optimal Transport". In: *Journal of Machine Learning Research* 22.78 (2021), pp. 1–8. URL: `http://jmlr.org/papers/v22/20-451.html`.

[21] Shivam Gautam et al. "SDVTracker: Real-time multi-sensor association and tracking for self-driving vehicles". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3012–3021.

[22] Andreas Geiger et al. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[23] Shengnan Guo et al. "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting". In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), pp. 3913–3926.

[24] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: `10.1162/neco.1997.9.8.1735`.

[25] S. Hoermann, M. Bach, and K. Dietmayer. "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling". In: *2018 IEEE International Conference On Robotics And Automation (ICRA)* (2018), pp. 2056–2063.

[26] Siyu Huang et al. "Deep learning driven visual path prediction from a single image". In: *IEEE Transactions on Image Processing* 25.12 (2016), pp. 5892–5904.

[27] M. Itkina, K. Driggs-Campbell, and M. Kochenderfer. "Dynamic environment prediction in urban scenes using recurrent representation learning". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), pp. 2052–2059.

[28] John Krafcik. *Waymo is opening its fully driverless service to the general public in Phoenix*. 2021. URL: `https://blog.waymo.com/2020/10/waymo-is-opening-its-fully-driverless.html`.

[29] J. Johnson, A. Alahi, and L. Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European Conference On Computer Vision* (2016), pp. 694–711.

[30] L. V. Kantorovich. "Mathematical Methods of Organizing and Planning Production". In: *Management Science* 6.4 (July 1960), pp. 366–422. DOI: `10.1287/mnsc.6.4.366`. URL: `https://doi.org/10.1287/mnsc.6.4.366`.

[31] Sertac Karaman and Emilio Frazzoli. "Incremental sampling-based algorithms for optimal motion planning". In: *Robotics Science and Systems VI* 104.2 (2010).

[32] Christoph G Keller and Dariu M Gavrila. "Will the pedestrian cross? a study on pedestrian path prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 15.2 (2013), pp. 494–506.

[33] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[34] Julian Francisco Pieter Kooij et al. "Context-based pedestrian path prediction". In: *European Conference on Computer Vision*. Springer. 2014, pp. 618–633.

[35] Bernard Lange, Masha Itkina, and Mykel J Kochenderfer. "Attention Augmented ConvLSTM for Environment Prediction". In: *arXiv preprint arXiv:2010.09662* (2020).

[36] Sam Levin and Julia Carrie Wong. "Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian". In: *The Guardian* (Mar. 19, 2018). URL: `https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe`.

[37] Lingyun Luke Li et al. "End-to-end contextual perception and prediction with interaction transformer". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5784–5791.

[38] Ji Lin, Chuang Gan, and Song Han. "Tsm: Temporal shift module for efficient video understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7083–7093.

[39] William Lotter, Gabriel Kreiman, and David Cox. “Deep predictive coding networks for video prediction and unsupervised learning”. In: *arXiv preprint arXiv:1605.08104* (2016).

[40] Abdelhak Loukkal et al. “Driving among Flatmobiles: Bird-Eye-View occupancy grids from a monocular camera for holistic trajectory planning”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 51–60.

[41] Chenyang Lu, Marinus Jacobus Gerardus van de Molengraft, and Gijs Dubbelman. “Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 445–452.

[42] Wenjie Luo, Bin Yang, and Raquel Urtasun. “Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 3569–3577.

[43] Yuexin Ma et al. “Trafficpredict: Trajectory prediction for heterogeneous traffic-agents”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 6120–6127.

[44] N. Mohajerin and M. Rohani. “Multi-step prediction of occupancy grid maps with recurrent neural networks”. In: *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition* (2019), pp. 10600–10608.

[45] Julien Moras, Véronique Cherfaoui, and Philippe Bonnifait. “Evidential grids information management in dynamic environments”. In: *17th International Conference on Information Fusion (FUSION)*. IEEE. 2014, pp. 1–7.

[46] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

[47] Dominik Nuss et al. “A random finite set approach for dynamic occupancy grid maps with real-time application”. In: *The International Journal of Robotics Research* 37.8 (2018), pp. 841–866.

[48] Peter Ondruska and Ingmar Posner. “Deep tracking: Seeing beyond seeing using recurrent neural networks”. In: *Thirtieth AAAI conference on artificial intelligence*. 2016.

[49] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. “A survey on performance metrics for object-detection algorithms”. In: *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE. 2020, pp. 237–242.

[50] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[51] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[52] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[53] Ewoud AI Pool, Julian FP Kooij, and Dariu M Gavrila. “Using road topology to improve cyclist path prediction”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 289–296.

[54] Edgar Riba et al. *Kornia: an Open Source Differentiable Computer Vision Library for PyTorch*. Oct. 2019.

[55] Thomas Roddick and Roberto Cipolla. “Predicting semantic map representations from images using pyramid occupancy networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11138–11147.

[56] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).

[57] M. Schreiber, S. Hoermann, and K. Dietmayer. “Long-term occupancy grid prediction using recurrent neural networks”. In: *2019 International Conference On Robotics And Automation (ICRA)* (2019), pp. 9299–9305.

[58] Marcel Schreiber et al. “A Multi-Task Recurrent Neural Network for End-to-End Dynamic Occupancy Grid Mapping”. In: *arXiv preprint arXiv:2202.04461* (2022).

[59] Glenn Shafer. “Dempster-shafer theory”. In: *Encyclopedia of artificial intelligence* 1 (1992), pp. 330–331.

[60] P. Sun et al. “in perception for autonomous driving: Waymo open dataset”. In: *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition* (2020), pp. 2446–2454.

[61] Maneekwan Toyungyernsub et al. “Double-prong convlstm for spatiotemporal occupancy prediction in dynamic environments”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13931–13937.

[62] Yunbo Wang et al. “Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5123–5132.

[63] Z. Wang, E. Simoncelli, and A. Bovik. “Multiscale structural similarity for image quality assessment”. In: *The Thrity-Seventh Asilomar Conference On Signals, Systems & Computers, 2003* 2 (2003), pp. 1398–1402.

[64] Z. Wang et al. “from error visibility to structural similarity”. In: *IEEE Transactions On Image Processing* 13 (2004), pp. 600–612.

[65] World Health Organization. *Road traffic injuries*. 2021. URL: `https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries`.

[66] P. Wu, S. Chen, and D. MotionNet Metaxas. “Joint Perception and Motion Prediction for Autonomous Driving Based on Bird’s Eye View Maps”. In: *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition* (2020), pp. 11385–11395.

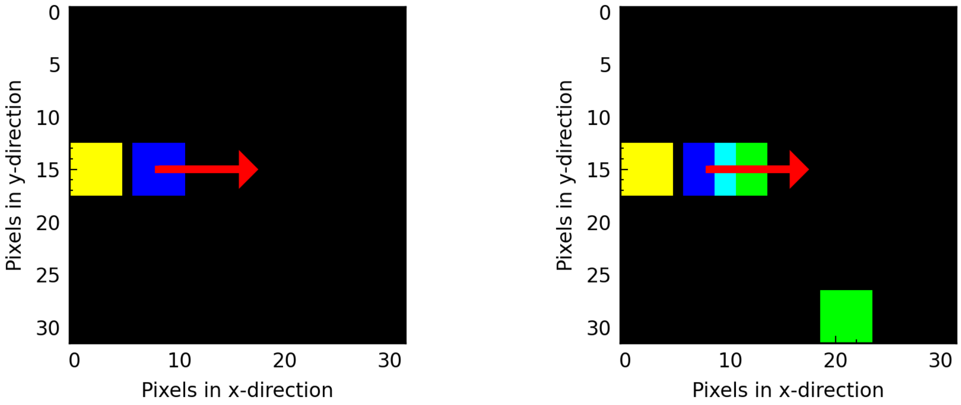
[67] SHI Xingjian et al. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting”. In: *Advances in neural information processing systems*. 2015, pp. 802–810.

[68] Hui Xiong et al. “Recurrent neural network architectures for vulnerable road user trajectory prediction”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 171–178.

A

Hyperparameter results of the sliding square toy experiment

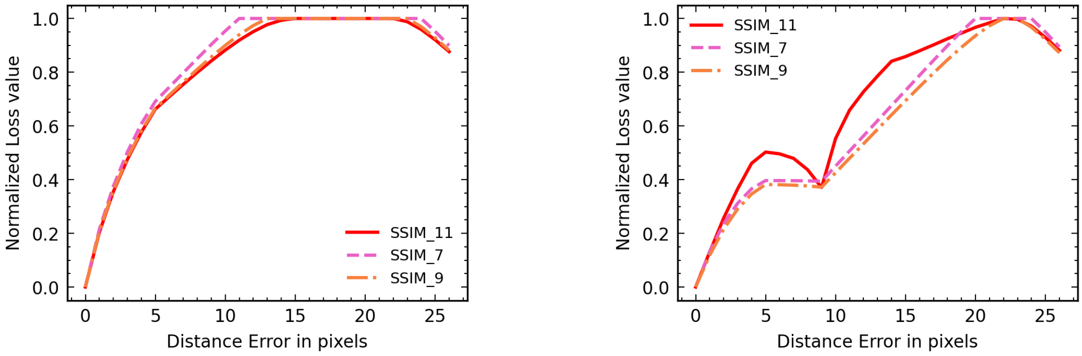
Figures A.2 and A.3 show the results for the two experiment scenarios in figure A.1 from the sliding square toy experiment described in section 4.1.1.



(a) The first simulated scenario.

(b) The second simulated scenario.

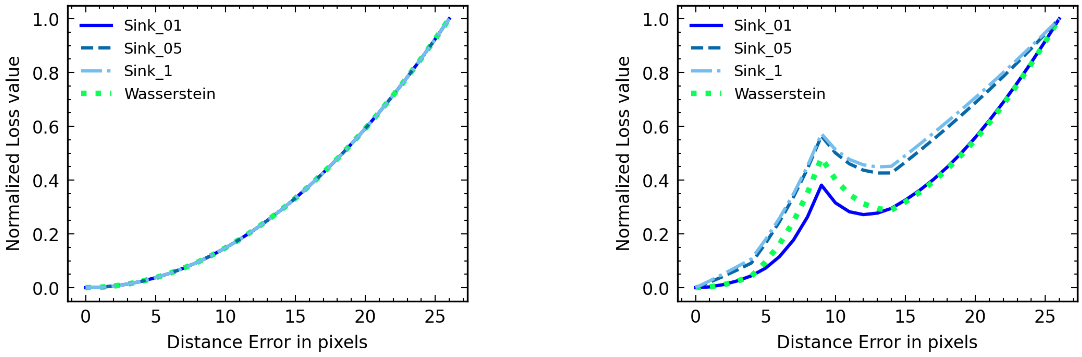
Figure A.1: The two scenarios that are generated to investigate the effect of the different loss functions on an increasing distance error. Yellow: Ground truth / False negative prediction, Green: True positive prediction, Blue: The moving square in the simulated predictions, Cyan: Overlap of the moving square and true positive predictions, Red arrow: The direction in which the moving square slides, Black: Empty.



(a) The SSIM loss hyperparameter results for the first scenario.

(b) The SSIM loss hyperparameter results for the second scenario.

Figure A.2: The normalized graphs for the SSIM loss with the different window sizes 11x11 (SSIM_11), 9x9 (SSIM_9), and 7x7 (SSIM_7) for the two scenarios where the ground truths are compared to predictions with increasing distance errors.



(a) The Sinkhorn loss hyperparameter results for the first scenario.

(b) The Sinkhorn loss hyperparameter results for the second scenario.

Figure A.3: The normalized graphs for the Sinkhorn loss with the different blur values 0.1 (Sink_01), 0.5 (Sink_05), and 1 (Sink_1) for the two scenarios where the ground truths are compared to predictions with increasing distance errors.

B

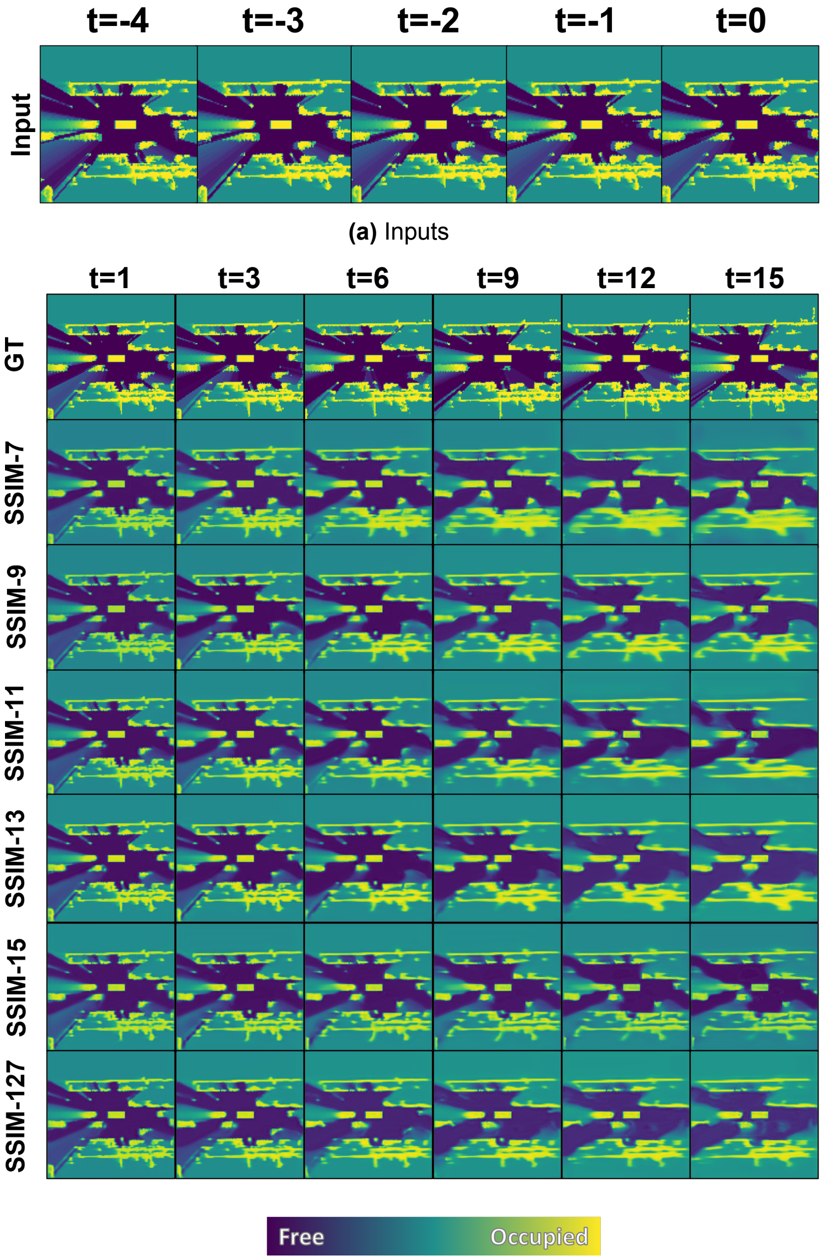
Hyperparameter search study for the SSIM, Sinkhorn and Sinkhorn-L1 loss functions

B.1. Loss hyperparameter search study

A hyperparameter search study is performed to investigate the influence of the SSIM and Sinkhorn losses' window size and blur value parameters respectively on the OGM prediction task using the PredRNN++ network. First the SSIM loss is investigated for different window sizes. This is followed by an investigation on the Sinkhorn loss and the Sinkhorn-L1 composite loss with different blur values for the Sinkhorn loss.

B.1.1. SSIM window size hyperparameter search

The OGM prediction experiments using the PredRNN++ network, as described in section 4.2, are repeated using the SSIM loss for different window sizes. The window sizes with equal sides of 7 to 15 pixels with intervals of 2 are investigated, as well as the window size of 127x127 which is the largest possible size to slide over the 128x128 OGM, almost spanning the whole OGM. The qualitative results are shown in figure B.1 and the quantitative results for the MSE, IS, AP, and Accuracy metrics are shown in table B.1. When looking at the quantitative results, the SSIM with a window size of 9x9 performs best on the MSE and Accuracy metric for both $T = 5$ and $T = 15$. For the IS and AP metrics, the SSIM with window size 127x127 performs best for $T = 5$, while the best scores for $T = 15$ are achieved by the window sizes of 7x7 and again 127x127 respectively. Based on both qualitative and quantitative results, there is no trend between the window size and prediction performance. This would mean that the window size should be selected empirically. The qualitative results show that the SSIM with a window size of 15x15 predicts the least blur and least object disappearances compared to the SSIM predictions with other window sizes. However, the movement of the overtaking vehicle is best captured by the 13x13 window predictions.



(a) Inputs

(b) Outputs

Figure B.1: The qualitative results on a Waymo Open Perception test sequence for PredRNN++ trained with the SSIM loss function with different window sizes. Results are shown for $T = 15$. (a) The five input frames to the network, showing three vehicles approaching the ego-vehicle (central yellow rectangle) from behind (left of image), and three vehicles moving away from the ego-vehicle (right). (b) Ground Truth (GT) and predictions for the loss functions for selected timesteps. For the window sizes of 7x7 and 15x15, the occupancy of the vehicles at the right side of the image are the least blurred and disappeared. For the window size of 13x13, the overtaking manoeuvre of the vehicle closest to the ego-vehicle in the input image at t=0, is best predicted without blurring. The SSIM with window size 15x15 shows the least blurring on the static objects and the free space.

Table B.1: The quantitative comparison on the Waymo Perception test set for PredRNN++ trained with the SSIM loss function using different window sizes. Performance is measured using the MSE and IS metrics (lower is better) and AP and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. The SSIM with a window size of 9x9 performs best on the MSE and Accuracy metrics for both prediction horizons. The SSIM with a window size of 127x127 performs best on the IS and for $T = 5$ and the AP scores for both prediction horizons. For $T = 15$ the best IS score is achieved with the SSIM with window size 7x7.

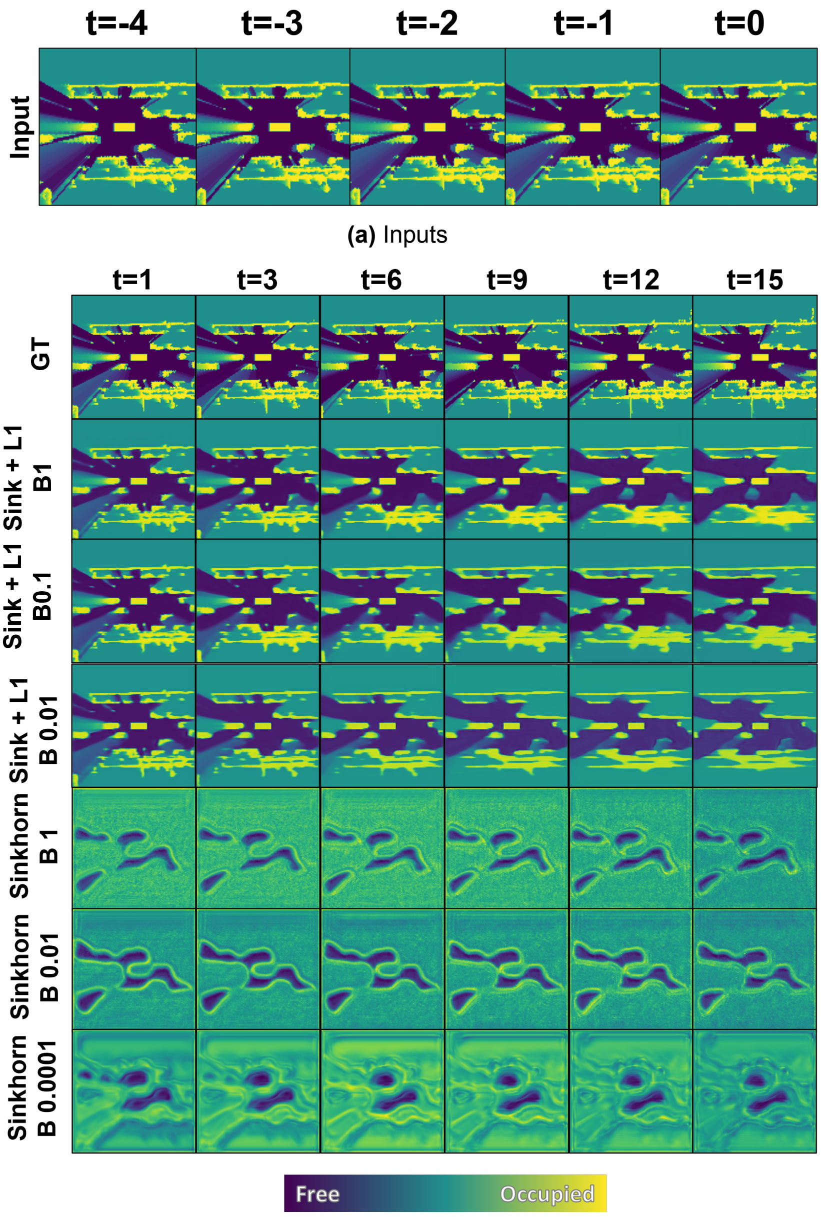
Loss Function	**Hyperparameters**	**MSE (x10^{-2}) ↓**	**IS ↓**	**AP ↑**	**Accuracy ↑**
T=5					
SSIM	N=M=7	2.22 ±1.92	1.370 ±1.537	0.942 ±0.055	0.882 ±0.130
SSIM	N=M=9	**1.57 ±1.10**	0.949 ±0.716	0.951 ±0.049	**0.923 ±0.046**
SSIM	N=M=11	1.76 ±1.22	1.009 ±0.761	0.949 ±0.051	0.916 ±0.047
SSIM	N=M=13	1.76 ±1.16	1.007 ±0.682	0.951 ±0.049	0.916 ±0.046
SSIM	N=M=15	1.71 ±1.17	1.025 ±0.772	0.948 ±0.052	0.915 ±0.048
SSIM	N=M=127	1.78 ±1.49	**0.947 ±0.759**	**0.952 ±0.050**	0.914 ±0.065
T=15					
SSIM	N=M=7	4.03 ±3.03	**1.985 ±1.499**	0.893 ±0.105	0.846 ±0.114
SSIM	N=M=9	**3.21 ±2.40**	2.091 ±1.178	0.905 ±0.104	**0.867 ±0.086**
SSIM	N=M=11	4.48 ±3.74	2.920 ±2.742	0.912 ±0.102	0.835 ±0.114
SSIM	N=M=13	3.76 ±2.85	2.033 ±1.592	0.907 ±0.104	0.865 ±0.080
SSIM	N=M=15	3.73 ±2.82	2.250 ±1.851	0.900 ±0.109	0.860 ±0.086
SSIM	N=M=127	3.59 ±2.82	2.097 ±1.845	**0.914 ±0.099**	0.866 ±0.086

B.1.2. Sinkhorn blur value hyperparameter search

This hyperparameter search study also repeats the OGM prediction experiments using the PredRNN++ network, as described in section 4.2. This experiment looks at the network's behavior using the Sinkhorn-L1 composite loss for the blur values 1, 0.1 and 0.01. Moreover, it investigates the blur values of 1, 0.01, and 0.0001 for the Sinkhorn loss. The qualitative results are shown in figure B.2 and the quantitative results for the MSE, IS, AP, and Accuracy metrics are shown in table B.2. The quantitative results show that the Sinkhorn-L1 composite loss performs best except for the AP for both prediction horizons and the IS at $T = 15$. The Sinkhorn-L1 with a blur of 1 performs best on the IS and AP at $T = 15$, while the one with a blur of 0.01 performs best on the AP at $T = 5$ and equally well for the Accuracy at $T = 15$ compared to the Sinkhorn-L1 with a blur of 0.1. The Sinkhorn losses all perform worse than the Sinkhorn-L1 composite losses. Among the Sinkhorn losses, the Sinkhorn with a blur of 1 outperforms or equals the other blur values for all metrics except the IS. For the IS, the Sinkhorn with a blur of 0.0001 performs best at $T = 5$ while the Sinkhorn with a blur of 0.01 performs best at $T = 15$. For the Sinkhorn loss at both prediction horizons, the AP metric shows a trend that for higher blur values, the AP value increases, thus improves. However, for the Sinkhorn-L1 loss, no such trend is seen. The qualitative results show that the Sinkhorn-L1 composite loss with a blur value of 0.01 has fewer disappearances of the vehicles for the long term predictions compared to the blur values of 0.1 and 1. Qualitatively, the Sinkhorn loss does not perform well for all blur values and only shows approximate areas of the free space locations in the ground truth. The higher the blur value, the less contrasting details are predicted and the more the predictions converge to an overall uncertain map.

Table B.2: The quantitative comparison on the Waymo Perception test set for PredRNN++ trained with the Sinkhorn loss and Sinkhorn-L1 composite loss for different blur values. Performance is measured using the MSE and IS metrics (lower is better) and AP and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. The Sinkhorn-L1 composite with a blur value of 0.1 performs best or equally for most metrics except for the AP for both prediction horizons and the IS at $T = 15$. The Sinkhorn-L1 with a blur of 1 performs best on the IS and AP metrics at $T = 15$. The Sinkhorn-L1 with a blur of 0.01 performs best for the AP at $T = 5$ and equally well for the Accuracy at $T = 5$ compared to the Sinkhorn-L1 with a blur of 1. The Sinkhorn losses do not perform well compared to the Sinkhorn-L1 composite losses. However, the Sinkhorn loss with a blur value of 1 performs best or equally good for most metrics except the IS compared to the Sinkhorn loss with other blur values. For the IS metric, among the Sinkhorn losses, the Sinkhorn with blur 0.0001 and 0.01 perform best for the time horizons $T = 5$ and $T = 15$ respectively.

Loss Function	**Hyperparameters**	**MSE (x10^{-2}) ↓**	**IS ↓**	**AP ↑**	**Accuracy ↑**
T=5					
Sinkhorn + L1	ϵ=1	2.12 ±1.38	1.129 ±0.716	0.943 ±0.053	0.911 ±0.042
Sinkhorn + L1	ϵ=0.1	**1.84 ±1.13**	**1.120 ±0.727**	0.944 ±0.052	**0.914 ±0.039**
Sinkhorn + L1	ϵ=0.01	2.07 ±1.23	1.175 ±0.702	**0.945 ±0.055**	0.909 ±0.039
Sinkhorn	ϵ=1	116.60 ±19.12	86.170 ±30.850	0.636 ±0.172	0.398 ±0.172
Sinkhorn	ϵ=0.01	229.79 ±34.59	15.801 ±43.009	0.595 ±0.159	0.398 ±0.172
Sinkhorn	ϵ=0.0001	218.00 ±37.85	9.240 ±33.789	0.476 ±0.139	0.398 ±0.172
T=15					
Sinkhorn + L1	ϵ=1	4.70 ±3.41	**2.179 ±1.535**	**0.904 ±0.095**	0.853 ±0.081
Sinkhorn + L1	ϵ=0.1	**3.75 ±2.61**	2.293 ±1.784	0.899 ±0.100	**0.860 ±0.080**
Sinkhorn + L1	ϵ=0.01	4.19 ±2.95	2.301 ±1.736	0.901 ±0.107	**0.860 ±0.075**
Sinkhorn	ϵ=1	112.44 ±19.02	46.311 ±40.296	0.615 ±0.174	0.397 ±0.172
Sinkhorn	ϵ=0.01	226.61 ±33.84	38.781 ±48.661	0.582 ±0.171	0.397 ±0.172
Sinkhorn	ϵ=0.0001	188.68 ±43.53	47.579 ±55.178	0.494 ±0.143	0.397 ±0.172



(a) Inputs

(b) Outputs

Figure B.2: The qualitative results on a Waymo Open Perception test sequence for PredRNN++ trained with the Sinkhorn loss and Sinkhorn-L1 composite loss for different blur values. Results are shown for $T = 15$. (a) The five input frames to the network, showing three vehicles approaching the ego-vehicle (central yellow rectangle) from behind (left of image), and three vehicles moving away from the ego-vehicle (right). (b) Ground Truth (GT) and predictions for the loss functions for selected timesteps. The results for the Sinkhorn-L1 composite loss with blur values 1, 0.1, and 0.01 are similar. However, the loss with a blur value of 0.01 (B0.01) retains the six moving vehicles in the scene for a longer time horizon compared to the losses with a blur value of 0.1 (B0.1) and 1 (B1). The performances for all three blur values for the Sinkhorn loss show that the Sinkhorn loss alone does not provide perceptually similar OGMs compared to the ground truth or the Sinkhorn-L1 composite loss. One can see that the overal regions of free space are best represented by the Sinkhorn with a blur of 0.01. The structure of the static environment is best resembled by the Sinkhorn with a blur of 0.0001.

C
Semantic segmentation results from the multi-task learning network

The semantic segmentation results from the multi-task learning experiment of section 4.3 are described in this appendix. Initially, the experiment investigates the effect of multi-task learning the OGM prediction task and the semantic segmentation task on the performance of the OGM prediction task using the PredRNN++ [62] network. The performance of the semantic segmentation task is therefore less relevant for the conclusion of this thesis.

C.1. Results semantic segmentation

Two validation set sequences with their semantic channel predictions combined into semantic map RGB-images are shown in figures C.1 and C.2. Tables C.1, C.2, and C.3 show the quantitative prediction performances for each semantic channel where the network is trained by the L1, SSIM, and Sinkhorn-L1 losses respectively. The quantitative results for the L1 loss show that the free space and the unknown classes are predicted with high AP values for both prediction horizons compared to the other classes. The vehicle and static objects classes score average in the AP metric, while the pedestrians and cyclists have low AP values. When looking at the MSE, IS, and Accuracy metrics, the cyclists class has the best scores, with the pedestrians at a close second. However, the MSE, IS, and Accuracy metric scores can be misleading for evaluating the correctly predicted labels (true positives) because these metrics do not adjust the values according to the occurrence of positive class instances in the scenes. Vehicles, pedestrians, cyclists, and static objects cover a minority of the cells in most OGM scenes. So even if the network predicts no instances of these classes, the metric scores will still be good because it will result in a good true negative rate (i.e. correctly predicted cells without these class labels, which is a majority of the cells). Therefore, these metrics will only provide useful insights into the prediction quality if the corresponding AP score is high. From the qualitative results for the L1 loss, it can be seen that the free space, unknown space, and static occupied space classes are predicted accurately. The vehicle class is only predicted accurately for dynamic vehicles close to the ego-vehicle. The static vehicles or vehicles farther away from the ego-vehicles, together with the pedestrian and cyclist classes, are often not accurately predicted and are predicted as static objects instead. The quantitative results for the SSIM loss show similar results to the ones of the L1 loss. The free space and unknown classes have relatively high AP values compared to the other classes. The other classes have low AP values and good MSE and Accuracy scores, which do not mean much given the low AP values. Especially compared to the L1 loss predictions, the SSIM's vehicles and static object classes are predicted worse when looking at the AP values. The qualitative results for the SSIM also show that the free space and unknown areas are predicted well. The vehicles and static objects only show the outlines of the objects which can explain the worse quantitative performance compared to the L1 loss. Pedestrians and cyclists are not predicted at all. For the Sinkhorn-L1 loss, the quantitative results are similar to the SSIM, also considering the low AP values for the vehicles and static objects. The Sinkhorn-L1 loss shows the worst overall AP and MSE scores compared to the other losses. This is also reflected in the qualitative results. Free and unknown space are predicted in similar areas compared to the ground truth. However, other classes are not predicted by the Sinkhorn-L1 network. Both the SSIM and the Sinkhorn-L1 results show that the static object labels become lighter for an increasing prediction horizon. Perhaps the semantic label values are influenced by the evidential OGM values since those also become lighter due to increased uncertainty over time. The IS scores for the same classes are equal or similar in value when comparing the results from different losses. The IS metric can reach a maximum (distance) value if no instances are present in the ground truth but are present in the predictions. This can often be the case for vehicles, pedestrians, and cyclists since the Waymo dataset does not contain many scenes with these objects.

Table C.1: The quantitative results of predicting the semantic channels on the Waymo Perception validation set for PredRNN++. Performance is measured using the MSE and IS metrics (lower is better) and AP, and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. * The values for the MSE, IS, and Accuracy can be misleading because of the unbalanced class representations for the different semantic classes.

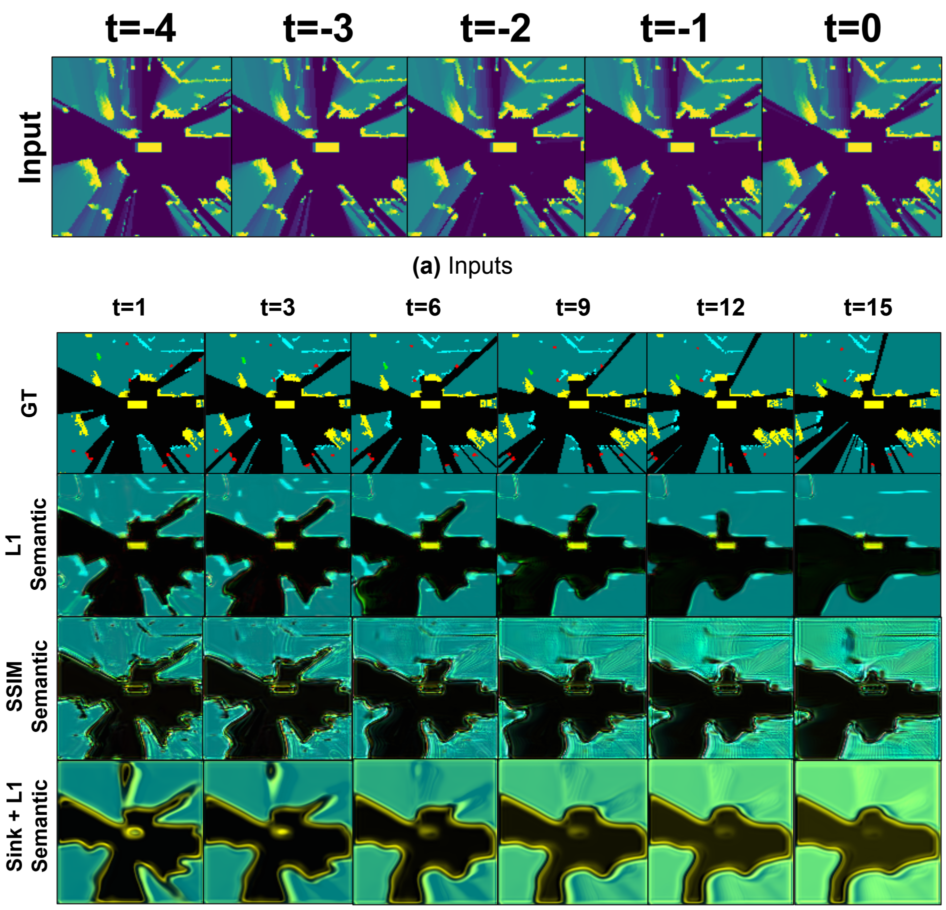
Channel	**MSE (x10^{-2})** ↓	**IS** ↓	**AP** ↑	**Accuracy** ↑
T=5		L1		
Free space	4.33 ±3.49	9.687 ±2.461	0.944 ±0.057	0.944 ±0.038
Vehicles	0.95 ±1.12	10.815 ±23.770	0.676 ±0.245	0.989 ±0.012
Pedestrians	0.02 ±0.07	3.135 ±23.126	0.006 ±0.009	1.000 ±0.001
Cyclists	0.02 ±0.05	1.132 ±14.589	0.014 ±0.035	1.000 ±0.000
Static objects	7.87 ±2.59	13.964 ±25.369	0.618 ±0.158	0.907 ±0.031
Unknown	7.82 ±3.73	5.282 ±18.714	0.934 ±0.061	0.901 ±0.040
T=15		L1		
Free space	7.24 ±6.02	10.999 ±2.982	0.895 ±0.093	0.916 ±0.062
Vehicles	0.89 ±1.10	9.873 ±17.915	0.667 ±0.210	0.988 ±0.012
Pedestrians	0.01 ±0.07	3.300 ±24.057	0.005 ±0.008	1.000 ±0.001
Cyclists	0.01 ±0.03	1.138 ±14.528	0.013 ±0.030	1.000 ±0.000
Static objects	8.28 ±2.77	13.989 ±25.614	0.454 ±0.220	0.907 ±0.031
Unknown	11.11 ±5.95	5.283 ±17.028	0.880 ±0.097	0.873 ±0.058

Table C.2: The quantitative results of predicting the semantic channels on the Waymo Perception validation set for PredRNN++. Performance is measured using the MSE and IS metrics (lower is better) and AP, and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. * The values for the MSE, IS, and Accuracy can be misleading because of the unbalanced class representations for the different semantic classes.

Channel	**MSE (x10^{-2})** $\downarrow$	**IS** $\downarrow$	**AP** $\uparrow$	**Accuracy** $\uparrow$
T=5	SSIM			
Free space	4.51 ±3.40	9.105 ±2.309	0.922 ±0.068	0.939 ±0.039
Vehicles	1.13 ±1.16	7.978 ±27.565	0.089 ±0.045	0.989 ±0.012
Pedestrians	0.01 ±0.07	3.135 ±23.126	0.005 ±0.006	1.000 ±0.001
Cyclists	0.02 ±0.03	1.132 ±14.589	0.009 ±0.020	1.000 ±0.000
Static objects	9.07 ±3.07	13.964 ±25.369	0.148 ±0.054	0.907 ±0.031
Unknown	8.48 ±3.87	5.158 ±18.970	0.914 ±0.077	0.865 ±0.046
T=15	SSIM			
Free space	7.16 ±5.64	10.135 ±2.786	0.863 ±0.108	0.911 ±0.064
Vehicles	1.15 ±1.16	8.238 ±28.260	0.049 ±0.050	0.989 ±0.012
Pedestrians	0.02 ±0.08	3.300 ±24.057	0.005 ±0.005	1.000 ±0.001
Cyclists	0.05 ±0.08	1.442 ±16.476	0.005 ±0.014	1.000 ±0.001
Static objects	9.08 ±3.06	13.989 ±25.614	0.131 ±0.052	0.907 ±0.031
Unknown	11.80 ±5.73	5.176 ±17.263	0.834 ±0.123	0.840 ±0.060

Table C.3: The quantitative results of predicting the semantic channels on the Waymo Perception validation set for PredRNN++. Performance is measured using the MSE and IS metrics (lower is better) and AP, and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$ with standard deviations. * The values for the MSE, IS, and Accuracy can be misleading because of the unbalanced class representations for the different semantic classes.

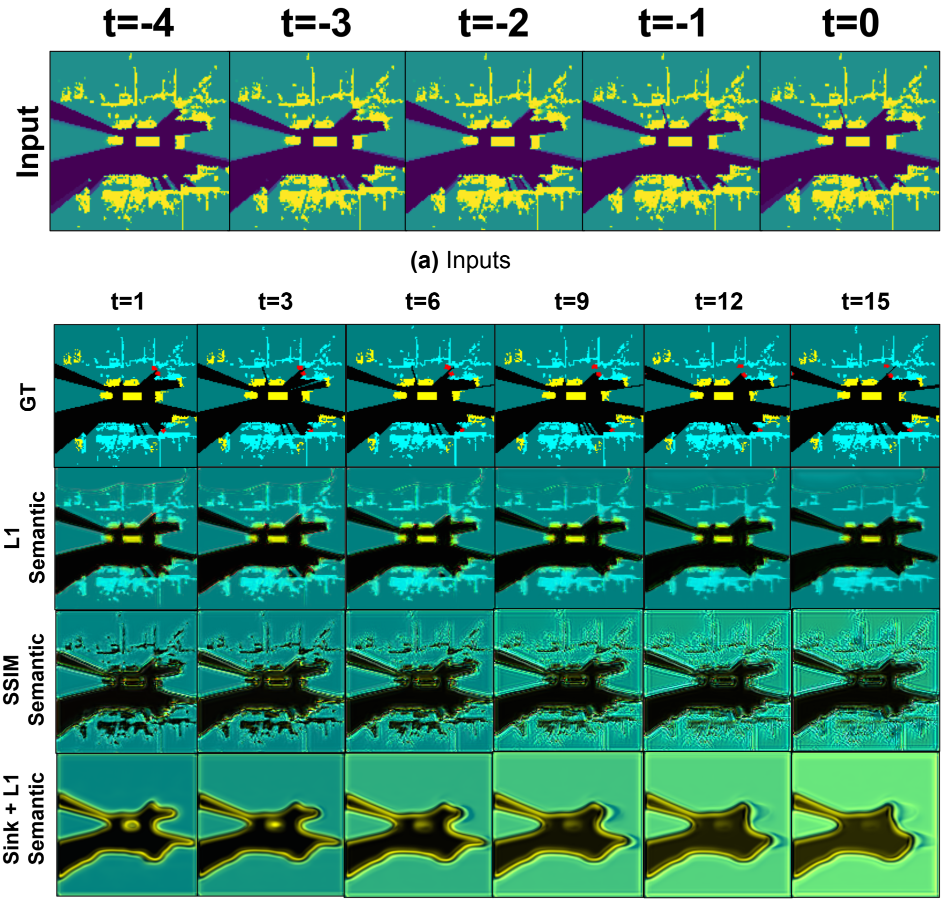
Channel	**MSE (x10^{-2})** $\downarrow$	**IS** $\downarrow$	**AP** $\uparrow$	**Accuracy** $\uparrow$
T=5	Sinkhorn + L1			
Free space	8.86 ±5.53	12.417 ±4.412	0.886 ±0.092	0.850 ±0.065
Vehicles	1.26 ±1.09	7.978 ±27.565	0.062 ±0.062	0.989 ±0.012
Pedestrians	0.54 ±0.28	3.135 ±23.126	0.004 ±0.004	1.000 ±0.001
Cyclists	1.85 ±0.37	1.132 ±14.589	0.002 ±0.002	1.000 ±0.000
Static objects	11.10 ±3.38	13.964 ±25.369	0.109 ±0.046	0.907 ±0.031
Unknown	14.19 ±5.01	5.968 ±18.164	0.840 ±0.084	0.823 ±0.054
T=15	Sinkhorn + L1			
Free space	14.86 ±9.85	15.985 ±5.975	0.799 ±0.141	0.815 ±0.083
Vehicles	1.38 ±1.11	9.476 ±32.240	0.042 ±0.050	0.989 ±0.012
Pedestrians	0.55 ±0.24	3.300 ±24.057	0.004 ±0.004	1.000 ±0.001
Cyclists	2.01 ±0.68	1.183 ±14.528	0.003 ±0.003	1.000 ±0.000
Static objects	11.03 ±3.42	13.989 ±25.614	0.103 ±0.042	0.907 ±0.031
Unknown	17.71 ±7.13	6.267 ±17.393	0.826 ±0.103	0.797 ±0.068



(a) Inputs

(b) Outputs

Figure C.1: The first example of qualitative results on a Waymo Open Perception validation sequence for PredRNN++ trained to perform semantic segmentation with different loss functions. Results are shown for $T = 15$. (a) The five input frames to the network show a crossing with four vehicles approaching the ego-vehicle (central yellow rectangle) from behind (left of image), one vehicle at the left side of the ego-vehicle (top of image), three vehicles in front of the ego-vehicle, of which two are closely behind each other (right), and a parked car (bottom right). Furthermore, at the bottom and top of the image are several pedestrians and there is one cyclists present at the top left in the image. (b) Ground Truth (GT) and the semantic predictions. In the semantic map, black: free, green-blue: uncertain, cyan: static occupied, yellow: vehicles, red: pedestrians, green: cyclists. The L1 loss semantic segmentation maps predict the overall free, unknown, and static classes similarly to the ground truth. However, only the ego-vehicle and one other vehicle (right above on the left to the ego-vehicle) are predicted for the vehicle class and no correct predictions are done for the pedestrian and cyclist classes. The cyclists, as well as some pedestrians are predicted as static objects for $T = 1$ and $T = 3$. They disappear for longer prediction horizons. The predictions show a thin green line around the unknown areas, however no cyclists are there. The SSIM loss results show free and unknown classes similarly to the ground truths. The static objects and vehicles classes only show outlines of the objects. No pedestrians or cyclists are predicted correctly. For a longer prediction horizon, the unknown class labels become lighter. The Sinkhorn-L1 results show free and unknown classes in a similar shape to the ground truth. At the edges between unknown and free space, an outline is predicted of the vehicle class. Also, the ego-vehicle is predicted, but disappears for longer a prediction horizon. Like the SSIM results, for a longer prediction horizon, the unknown class labels become lighter.



(a) Inputs

(b) Outputs

Figure C.2: The second example of qualitative results on a Waymo Open Perception validation sequence for PredRNN++ trained to perform semantic segmentation with different loss functions. Results are shown for $T = 15$. (a) The five input frames to the network show a road with four vehicles around the ego-vehicle (central yellow rectangle), one vehicle behind the ego-vehicle (left to ego-vehicle in image), two left of the ego-vehicle (above ego-vehicle in image), and one in front of the ego-vehicle (right to ego-vehicle in image). Two vehicles are further away in front of the ego-vehicle (right of the image). There are three parked cars (two at top left, one at bottom right in the image). Furthermore, there are pedestrians in front of the ego-vehicle at its left and right side (right top and bottom in the image). There are no cyclists in this scene. (b) Ground Truth (GT) and the semantic predictions. In the semantic map, black: free, green-blue: uncertain, cyan: static occupied, yellow: vehicles, red: pedestrians, green: cyclists. For the L1 loss, the semantic segmentation maps predict the overall free, unknown, and static classes similarly to the ground truth. The vehicles around the ego-vehicle are predicted correctly as vehicles for the full prediction horizon. The vehicles that are parked or are further away are predicted as static objects. No correct predictions are done for the pedestrian class. The pedestrians are predicted as static objects and blur over time. The predictions show a thin green line around the unknown areas, however no cyclists are there. The SSIM loss results show free and unknown classes similarly to the ground truths. The static objects and vehicles classes only show outlines of the objects and no pedestrians or cyclists are predicted correctly. For a longer prediction horizon, the unknown class labels become lighter. The Sinkhorn-L1 results show free and unknown classes in a similar shape to the ground truth. At the edges between unknown and free space, an outline is predicted of the vehicle class. Also, the ego-vehicle is predicted, but disappears for longer a prediction horizon. Like the SSIM results, for a longer prediction horizon, the unknown class labels become lighter.

D

Submitted article: An Experimental Study on Occupancy Grid Map Prediction Loss Functions

An Experimental Study on Occupancy Grid Map Prediction Loss Functions

Hidde J-H. Boekema, Rutger F. Dirks, and Dariu M. Gavrila[1]

Abstract— **This paper presents an experimental study on the loss functions used in the Occupancy Grid Map (OGM) prediction literature. We examine the ubiquitous L1 and L2 loss functions, and composites thereof, as well as the Binary Cross-Entropy (BCE) and distance-capturing SSIM losses in an OGM prediction setting.**

In experiments on the large-scale Waymo Open Perception dataset with the influential PredRNN++ video prediction network, we find that the L1 loss function generally achieves the best performance on a range of per-pixel and perceptual metrics such as Mean Squared Error (MSE), Image Similarity (IS), Average Precision (AP), and Accuracy. However, we find some evidence that the SSIM and composite L1L2 and SmoothL1 losses result in reduced blurriness and fewer disappearing dynamic objects - despite scoring less well on these commonly-used metrics - and may therefore be more suitable for safety-critical applications.

I. Introduction

Estimation of the future state of the environment is essential to the safe deployment of Autonomous Vehicles (AVs) in urban environments. The complex interactions between road users and varied road layouts in such environments make this an especially challenging problem.

The classic approach to this problem is to use a hand-designed detection and tracking pipeline to obtain the trajectories of objects in a scene, which are then used to predict their future states (e.g. [1], [2]). Using this pipeline ensures that predictions are made efficiently for all detected agents in the scene due to the sparsity of the object-centric representation. A drawback is the dependence on object detectors and trackers. These components are additional sources of error (e.g. the object detection recall of the Fast and Furious network [3] is 92.5%) that can impact the performance of the prediction network; in the extreme case, missed detections can result in agents being left out of predictions, posing a safety hazard for intelligent vehicles.

An alternative approach is to predict directly from minimally processed sensor data. Approaches using the Occupancy Grid Map (OGM) [4] environment representation fall into this category. 'Evidential' OGMs are discrete maps with cells containing belief mass for the occupancy state of the corresponding location in the environment. They can be generated from sensor data using Dempster-Shafer theory [5] and therefore preserve much of the information collected by the sensors. However, predicting on such a low-level representation of the environment is challenging. This is compounded by the use of 'per-pixel' loss functions, which ignore spatial dependencies between grid cells. [1] found that this led to the gradual disappearance of moving objects and blurry predictions over long prediction horizons.

This paper studies the effect of the loss function used for training on the performance of OGM prediction networks. The L1, L2, Binary Cross-Entropy, and SSIM loss functions, as well as combinations of the L1 and L2 losses, are evaluated on the large-scale Waymo Open Perception dataset [6]. We note that these result may be relevant outside of the domain of OGM prediction, for example in the similar video prediction domain.

II. Related Work

There is an extensive body of literature for the environment prediction problem. The existing work can be categorised into *object-centric* and *object-agnostic* prediction approaches. Object-centric approaches (e.g. see survey [7]) employ a pipeline consisting of object detection, localisation, and tracking to generate object-level input features for the prediction task. Recently, approaches using an object-agnostic representation for the prediction problem have gained popularity. This representation does not require a human-engineered pipeline but can be generated by minimally processing sensor data. We will focus on object-agnostic approaches in this section because of their relevance to our work. We will also discuss relevant works in the related video prediction problem.

Occupancy Grid Map Prediction As mentioned in Section I, objects become blurry or disappear over long prediction horizons [1]. This can be ascribed to the lack of semantic information in OGM predictions and to the use of per-pixel loss functions such as the L1 loss during training.

The loss functions commonly used in OGM prediction intersect with those for computer vision and image processing tasks because of the common format of images and OGMs. Image processing loss functions can be categorised as per-pixel loss and perceptual loss functions [8]. Per-pixel loss functions evaluate the differences between pixels of two images (or the grid cells of two OGMs), where each spatial location is evaluated in isolation to the rest of the image. These loss functions can estimate the similarity between corresponding pixels in two images efficiently, but cannot capture high-level image features as this requires consideration of the dependencies between pixels. Perceptual loss functions, on the other hand, measure the differences between high-level image features, resembling how the human visual system perceives differences [9].

Most state-of-the-art OGM prediction methods use per-pixel losses such as the L1, Mean Squared Error (MSE),

[1] Intelligent Vehicles Group, TU Delft, The Netherlands

and Cross Entropy losses [10], [1], [11], [12], [13], [14]. [1] use a perceptual metric to evaluate their OGM prediction results because a per-pixel metric, such as the Mean Squared Error (MSE), does not capture the positional variability of objects between the ground truth and a prediction. [15] use the Multi-scale Structural Similarity Index Metric (SSIM) [9], [16], a human visual perception-based loss function for training a part of their OGM prediction network, together with the Cross Entropy loss. [17] developed their own spatial and temporal consistency loss functions to train an OGM prediction network. These loss functions evaluate the occupancy, semantics and dynamics of the predictions, with the assumption that objects are rigid and move smoothly between consecutive frames. Moreover, the functions separately evaluate grid cells belonging to the same rigid object, allowing a prediction network to learn which grid cells belong to an object. However, their approach does not take into account the spatial distances between objects and other parts of the environment.

Hence, some research has been done towards developing a distance-based loss function suitable for training OGM prediction networks. However, current loss functions either do not capture spatial relations between objects, or are not evaluated in isolation ([15] combines the SSIM with the Cross Entropy loss). Loss functions that can measure dependencies between grid cells and hence distances between objects could improve the accuracy of OGM prediction networks trained with these loss functions.

Video Prediction The influential work by [18] introduced the Convolutional Long Short-Term Memory (ConvLSTM) network as an extension of Convolutional Neural Networks (CNNs) to sequences of images. Initially developed for precipitation forecasting, ConvLSTM uses past radar maps to predict a sequence of future maps. This model outperformed optical flow algorithms and Long Short-Term Memory models (LSTMs) [19] due to its ability to capture complex spatiotemporal patterns. However, it also produces increasingly blurry predictions for greater prediction horizons. [20] use ConvLSTM layers in their video prediction model, Convolutional Dynamic Neural Advection (CDNA). This model predicts pixel motions of frame segments and merges the segments into a single future frame prediction. Predictions made by CDNA also become blurrier over time due to the use of MSE loss in training, which causes uncertainty to be encoded as blur [20]. The authors suggest using an alternative loss to combat this issue.

PredRNN++ [21] mitigates the blurring of predictions by employing a 'gradient highway' that provides shorter routes for gradients to flow through a network. This allows the network to learn stronger spatial correlations and short-term dynamics, leading to more confident predictions and decreased blurriness over long prediction horizons. The PredNet [22] network has been used for OGM prediction [10], [1], [11] because of its desirable properties. Based on the neuroscientific concept of predictive coding, each layer of PredNet makes a local prediction using ConvLSTMs and forwards the errors of that prediction to the subsequent layers. The authors argue that this ensures that the network can learn an implicit model of the objects in the scenes, including their movement.

These approaches demonstrate that blurriness is in part caused by the increasing uncertainty of the model with time. This uncertainty could be reduced by ensuring more informative supervision signals are used in training through better loss functions

Our main contribution is documenting the performance differences resulting from the choice of loss function for OGM prediction - to our knowledge, such an experimental study has not been performed before. We note that our results are applicable to the video prediction domain as well.

III. Method

A. Problem Formulation

Environment prediction can be considered a self-supervised sequence-to-sequence prediction problem where a history sequence is used to predict a likely future sequence. We select evidential occupancy grid maps as our representation of the environment, as in [1], [11], [10], with OGMs generated using the Dempster-Shafer Theory (DST) [5], although our findings could generalise to other types of OGM. An evidential occupancy grid map of the environment $X_t \in \mathbb{R}_+^{H \times W \times 1}$ at time step t contains the occupancy (i.e. whether free or occupied) probability of each grid cell $x_{t,ij},\ i \in [1, H],\ j \in [1, W]$ at that instant. Given a past (observed) sequence of OGMs $X_{-\tau:0}$, the objective is to predict T frames into the future $X_{1:T}$.

B. Prediction Network

We use PredRNN++ [21] as the prediction network for our study. PredRNN++ is an influential method in the video prediction domain (for which it was developed), with an open-source implementation. It forms the basis of the OGM prediction network proposed by [1], achieving state-of-the-art results on the KITTI and Waymo Open Perception datasets.

C. Loss Functions

We consider six loss functions for OGM prediction in this work: the L1, L2, L1L2, SmoothL1, Binary Cross-Entropy (BCE), and SSIM [16] losses. The L1 loss is the most often used loss for training OGM prediction networks; the L2, or MSE, loss is an common alternative that penalises outliers more heavily. The combination of the L1 and L2 loss (which we call 'L1L2') is also tested as [21] claim that this loss simultaneously enhances the sharpness and smoothness of predicted frames in the video prediction task. We evaluate the BCE loss because [17] used this loss for the classification and state estimation heads of their proposed MotionNet prediction network. We also propose to study the SSIM function as a loss for this task due to its desirable features. Contrary to the losses in the literature, the SSIM metric considers dependencies between grid cells, and therefore penalises displacement errors in predictions.

The loss functions are aggregated over batches of N_b samples of $T_{loss} = T + \tau - 1$ timesteps as in Equation (1).

$$L = \frac{1}{N_b T_{loss}} \sum_{n=1}^{N_b} \sum_{t=1}^{T_{loss}} f(X_{t,n}, \hat{X}_{t,n}) \tag{1}$$

Where $f(\cdot, \cdot)$ is the applied loss function, and we denote a prediction by $\hat{X}$. Note that we drop the time (and sample) notation in the succeeding sections for clarity.

L1 loss: The L1 loss takes the mean of the absolute error between the grid cells of the ground truth X and prediction $\hat{X}$, as shown in Equation (2).

$$\text{L1}(X, \hat{X}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} |x_{ij} - \hat{x}_{ij}| \tag{2}$$

L2 loss: The L2 loss similarly takes the mean of the squared error between the grid cells of the ground truth and prediction (Equation (3)).

$$\text{L2}(X, \hat{X}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} |x_{ij} - \hat{x}_{ij}|^2 \tag{3}$$

Combined L1 and L2 ('L1L2') loss: We define the combined L1 and L2 loss of [21] as in Equation (4).

$$\text{L1L2}(X, \hat{X}) = \text{L1}(X, \hat{X}) + \lambda \cdot \text{L2}(X, \hat{X}) \tag{4}$$

λ is a constant weighting factor used to balance the effects of the L1 and L2 losses, with [21] selecting $\lambda = 1$.

Smooth L1 loss: Another variant on the L1 and L2 losses is the Huber [23] loss. It is a piecewise function that combines the advantages of the sensitivity of the mean-unbiased L2 function with the robustness of the median-unbiased L1 function. Equation (5) shows the Huber loss for two variables $a, b \in \mathbb{R}$.

$$\text{Huber}_\delta(a, b) = \begin{cases} \frac{1}{2}(a-b)^2 & \text{if } |a-b| < \delta, \\ \delta\left(|a-b| - \frac{1}{2}\delta\right) & \text{otherwise.} \end{cases} \tag{5}$$

Where δ determines the transition point between the losses. We use the Smooth L1 loss, a variant of the Huber loss that is used in [24], [17]. This loss is defined for the ground truth X and predicted $\hat{X}$ OGMs in Equation (6).

$$\text{SmoothL1}(X, \hat{X}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{1}{\delta} \text{Huber}_\delta(x_{ij}, \hat{x}_{ij}) \tag{6}$$

Binary Cross-Entropy (BCE): The cross-entropy loss is often used in logistic regression problems, such as semantic segmentation, and was proposed for OGM prediction by [17]. The BCE loss is a suitable candidate loss function for training evidential OGM prediction networks due to the similarity to logistic regression.

The loss is derived from the cross-entropy $H(\cdot, \cdot)$ of Bernoulli random variables $p \in \{y, 1-y\}$ and $q \in \{\hat{y}, 1-\hat{y}\}$ as in Equation (7).

$$\begin{aligned} H(p, q) &= -\sum_i p_i \log q_i \\ &= y \log(\hat{y}) - (1-y)\log(1-\hat{y}) \\ &= g(y, \hat{y}) \end{aligned} \tag{7}$$

Where we have introduced the notation $g(y, \hat{y})$ for convenience. The loss function takes the form of Equation (8) for OGM prediction.

$$\text{BCE}(X, \hat{X}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} g(x_{ij}, \hat{x}_{ij}) \tag{8}$$

SSIM loss: The Structural Similarity Index Measure (SSIM) [16] is a function inspired by the human visual system that measures the similarity between two images. It evaluates luminance, contrast, and structural information over windows of an image to estimate perceptual visual similarities [16]. The final loss value is computed by sliding the window over the image, calculating the SSIM locally, and taking the mean of the resultant values. The window size is a parameter that can be changed to adjust the scale for local evaluation. The components of the SSIM function are described in this section in the context of its application to OGM prediction.

Let the SSIM windows have size $N \times M$. The luminance μ_x of a window x of an OGM is the mean intensity over the window, which is calculated as in Equation (9). The luminance values of corresponding windows y and x of the ground truth and prediction, respectively, are compared using Equation (10), with the constant $C_1 = (K_1 L)^2$ stabilises the equation by preventing division by low values, where L is the dynamic range of the pixel intensity values, and $K_1 \ll 1$ is a small constant.

$$\mu_x = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \tag{9}$$

$$l(x, y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{10}$$

The contrast is the standard deviation σ_x of the window, shown in Equation (11). The contrasts of the windows of the ground truth and prediction are compared using Equation (12), with constant $C_2 = (K_2 L)^2$ where L is the dynamic range of the pixel intensity values and $K_2 \ll 1$ is a small constant again.

$$\sigma_x = \left(\frac{1}{NM - 1} \sum_{i=1}^{N} \sum_{j=1}^{M} (x_{ij} - \mu_x)^2 \right)^{1/2} \tag{11}$$

$$c(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{12}$$

Equation (13) is used to compare the structures, or correlations, between the prediction and the ground truth, where C_3 is a small constant and σ_{xy} is computed using Equation (14).

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{13}$$

$$\sigma_{xy} = \frac{1}{NM-1}\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{ij}-\mu_x)(y_{ij}-\mu_y) \tag{14}$$

After the comparisons are performed, they are combined according to Equation (15).

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{15}$$

$\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters to adjust the relative importance of the three factors. The SSIM function has a range of $[-1, 1]$, where a value of 1 means that the evaluated predicted window is identical to the ground truth one.

Since SSIM measures structural similarities, it penalises distance errors between OGMs because such errors change the structure of an image.

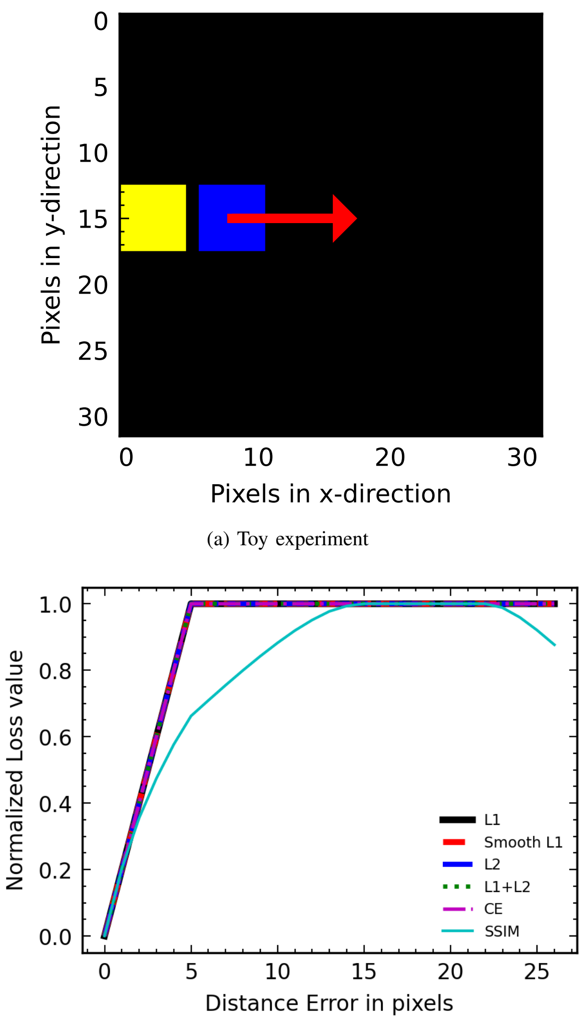
IV. Experiments

We first investigate the behaviour of the L1, L2, L1L2, SmoothL1, BCE, and SSIM losses in a toy experiment. We then train the PredRNN++ network [21] using each of the loss functions and report the performance on the test set of the real-world Waymo Perception dataset [6]. The experiments demonstrate the influence of the loss functions on the blurring of objects in the OGM predictions over a range of prediction horizons. Finally, the impact of the total training set on the difference in performance between the loss functions is studied.

We use the Torch Geometry library's [25] SSIM implementation. PyTorch [26] was used as deep learning framework, and for the L1, L2, L1L2, SmoothL1, and BCE loss function implementations. A Tesla V100-SXM2-32GB GPU is used for training and inference.

A. Toy Data

The L1, L2, L1L2, SmoothL1, BCE, and SSIM losses are compared in a toy experiment designed to investigate how the values of the losses vary with an increasing distance between two objects, like in [27]. A 32x32 ground truth image is created which contains a solid 5x5 square located at the center left of the image. Example predictions are generated which contain an identical square located at a range of horizontal offsets relative to the GT position ('distance errors'); see Figure 1a. The values of each loss function are recorded for the distance errors, then normalised to lie in the range $[0, 1]$ to aid comparison of the losses. The results of this evaluation are shown in Figure 1b. The graph shows that for distance errors greater than 5 pixels, where there is no overlap between the predicted and GT squares, the L1, L2, L1L2, SmoothL1, and BCE losses are constant. Although the SSIM loss initially increases with the distance error, it remains constant when a threshold distance error (dependent on the chosen window size) is reached; the SSIM loss also decreases when the prediction approaches the boundary of the image. This suggests that the SSIM loss is a more informative loss for OGM prediction networks.



(a) Toy experiment

(b) Normalised loss values

Fig. 1: Toy experiment to show the behaviour of the L1, L2, L1L2, SmoothL1, BCE, and SSIM losses. (a) A 'predicted' square (blue) is shifted away from the 'GT' square (yellow) along the x-axis in an image. (b) Curves showing the associated normalised values of the loss functions.

B. Waymo Open Perception Data

Experimental Setup: We evaluate the effect of training with each of the L1, L2, L1L2, SmoothL1, BCE, and SSIM losses on the test-time performance PredRNN++ prediction network [21] on the Waymo Open Perception dataset [6]. The Perception dataset contains 1150 unique scenes, recorded at a frame rate of 10 Hz. It contains millions of objects with tracking IDs, labels for four object classes (vehicles, pedestrians, cyclists, and signs), and 3D bounding boxes for each of those objects. We used the code of [11] to generate OGMs from the LiDAR data of the Perception dataset. The object class labels were projected into the OGM to generate the ground truth semantic maps. We obtain a dataset of 10000 training, 473 validation, and 2972 testing non-overlapping sequences of 10 frames. Additionally, we create 1405 longer test sequences of 20 frames from this dataset to test the trained networks for a longer prediction horizon of 1.5s.

The PredRNN++ [21] architecture is used for our experiments. The network consists of 4 blocks of causal ConvLSTMs with 12 layers each, with a gradient highway unit over the bottom LSTM layer. A 5x5 convolutional filter size is selected for the ConvLSTM units. We train with each of the losses on 32 mini-batches of 16 samples of the training set per epoch, for 200 epochs. The Adam optimiser [28] is used with a learning rate of $1\mathrm{e}{-3}$ and an exponential learning rate scheduler that updates the loss by a factor of 0.977 after every epoch. The network is trained to predict five future frames, given five past frames. We evaluate the trained network on the test set for a prediction horizon of five or 15 future frames.

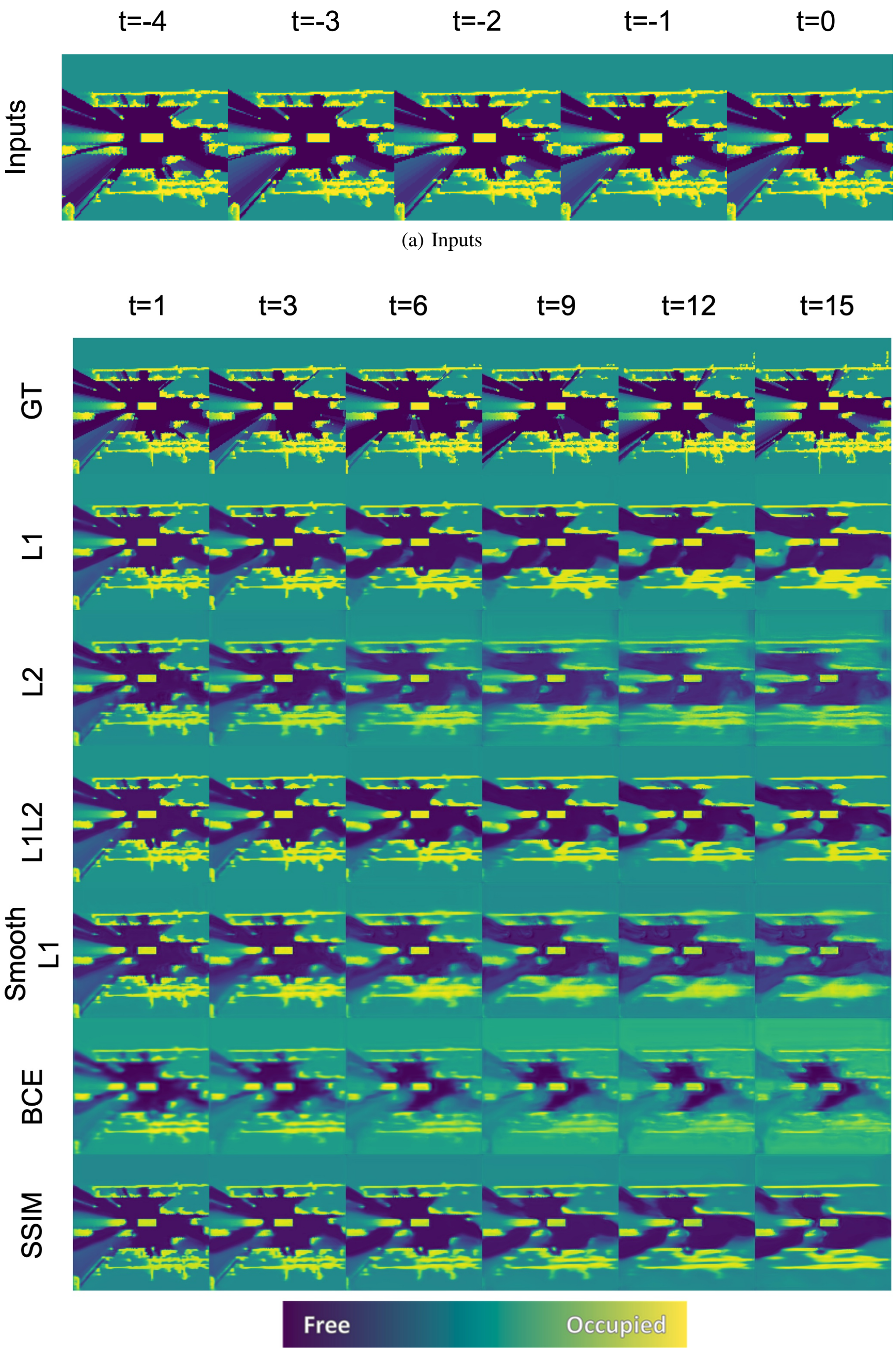
Evaluation Metrics: To evaluate our results, we use the Mean Squared Error (MSE) (as in [10], [1], [11]), Image Similarity (IS) [29] (as in [1]), Average Precision (AP), and Accuracy metrics [30]. The MSE metric measures the per-grid cell difference between the ground truth and predicted OGMs. IS is a perceptual metric that accounts for difference in scene structure between the ground truth and predictions by computing a distance map for grid cells with the same occupancy value, then averaging over these values. The occupancy values for evidential grids can be categorised as free (0), unknown (0.5), and occupied (1). This metric penalises blurriness in predictions in this setup, since blurred (predicted) cells (i.e. with value around 0.5) will belong to a different category than the corresponding ground truth cell and warp the computed distance maps as a result. The AP and Accuracy metrics are often employed for classification tasks; we use them to assess the effectiveness of our method in retaining the occupancy class of grid cells.

Results: We list the performance of the various loss functions in Table I for prediction horizons of $T = 5$ and $T = 15$. It can be seen that the performance of the network trained on a specific loss is dependent on the metric used to quantify the prediction error. The L1 loss function outperforms all other losses on the majority of the metrics across the prediction horizons, and scores highly on the remainder of the metrics. This is especially pronounced for the long prediction horizon, where it achieves a 14.4% lower MSE than L2, the next closest loss. Surprisingly, it even scores better on the perceptual IS metric than the SSIM loss (by a margin of 8.02% for $T = 15$), which optimises an objective function similar to the IS metric and explicitly captures spatial dependencies. An explanation for this phenomenon is that the L1 loss provides more robust and informative gradients to the network at train time than the SSIM loss. The composite L1L2 and SmoothL1 losses lie between the L1 and L2 losses on most metrics. For $T = 15$, however, these losses are generally worse than either loss alone, indicating that they do not combine the advantages of these losses, as claimed by [21]. The BCE loss appears least suitable for OGM prediction, as it scores poorly on all of the metrics.

Predictions on an example sequence are shown in Figure 2. Figure 2a shows the evidential OGMs input to the network, with yellow pixels representing grid cells with high probability of being occupied. The rectangle in the centre of the images represents the ego-vehicle. There are several vehicles on the road (dark purple) with the ego-vehicle. It can be seen that the the L1 loss produces sharp predictions, but that most of the vehicles disappear over even short prediction horizons - the vehicles below and to the right of the ego-vehicle are not predicted at $t = 9$. This could lead to critical safety issues in real-world deployment of the prediction system. This also demonstrates that the L1 loss is able to minimise the metrics by accurately predicting 'easy' examples (e.g. the empty road and static objects), and may therefore not be optimal for motion planners using the output of the prediction module. The predictions from the L2-trained network are blurrier, showing greater uncertainty, but better predict other road users than with L1. The L1L2 and SmoothL1 losses combine the advantages of the L1 and L2 losses, producing more certain predictions than L2 while maintaining vehicles in the future predictions. Specifically, the predictions from the L1L2 network are confident and have probability mass for all road users over the entire horizon, while the SmoothL1 network more accurately predicts the future position of the moving objects (at the expense of lower certainty). The BCE loss causes the most blurry predictions, as well as poor state estimation. The scene structure is maintained in the SSIM predictions, and the vehicles around the ego-vehicle are accurately predicted, although the small objects to the right of the OGM disappear after $t = 6$. This may be due to the window size of the SSIM, which affects the scale of objects that the loss function accounts for. Tuning this hyperparameter or using a multi-scale variant of the SSIM loss may mitigate this issue.

V. Conclusion

In this paper, we presented an experimental study on the loss functions proposed in the OGM prediction literature to mitigate well-documented issues like the disappearance of objects over long prediction horizons and blurring of predictions. We investigated popular losses such as L1 and L2 (and combinations thereof), as well as other losses from the literature i.e. Binary Cross-Entropy and SSIM. We found that training with the L1 loss function leads to the most accurate per-pixel predictions, following expectations, but also outperforms the perceptual SSIM loss on the perceptual IS metric. However, we note from qualitative examples that dynamic objects are prone to disappearing in predictions with this loss function, impacting the usefulness of this loss function for safety-critical applications such as environment prediction for Autonomous Vehicles (AVs). There is also some evidence in the form of qualitative examples that alternative losses like the SSIM and L1L2 losses combat blurriness in predictions, although this needs to be confirmed with further investigation. Future work also includes extending this study to other OGM types, such as semantic OGMs.



(a) Inputs

(b) Outputs

Fig. 2: Qualitative results on a Waymo Open Perception test sequence for PredRNN++ trained with the L1, L2, L1L2, SmoothL1, BCE, and SSIM loss functions. Results are shown for $T = 15$. (a) The five input frames to the network, showing three vehicles approaching the ego-vehicle (central yellow rectangle) from behind (left of image), and three vehicles moving away from the ego-vehicle (right). (b) Ground Truth (GT) and predictions for the loss functions for selected timesteps. The per-pixel losses lead to blurrier predictions than SSIM over long time horizons.

TABLE I: Quantitative comparison on the Waymo Perception test set for PredRNN++ trained with various loss functions. Performance is measured using the MSE and IS metrics (lower is better) and AP and Accuracy metrics (higher is better), averaged over prediction horizons $T = 5$ and $T = 15$. L1 excels on the majority of the metrics, also scoring highly on the remaining metrics.

		$\mathbf{T = 5}$				$\mathbf{T = 15}$			
Loss Function	**Hyperparameters**	**MSE (x10^{-2}) ↓**	**IS ↓**	**AP ↑**	**Accuracy ↑**	**MSE (x10^{-2}) ↓**	**IS ↓**	**AP ↑**	**Accuracy ↑**
L1	–	2.63	**1.830**	0.914	**0.892**	**3.60**	2.743	0.874	**0.854**
L2	–	2.75	2.306	0.923	0.870	4.20	**2.697**	**0.880**	0.817
L1L2	$\lambda = 1$	**2.60**	2.004	**0.936**	0.885	4.53	3.468	0.866	0.833
SmoothL1	$\delta = 1$	2.84	2.333	0.924	0.876	4.50	3.339	0.856	0.835
BCE	–	168.30	30.937	0.845	0.596	451.87	33.228	0.788	0.568
SSIM	$N = M = 11$	2.97	1.852	0.911	0.883	4.63	2.932	0.866	0.839

Acknowledgment

This work is part of the research programme Efficient Deep Learning (EDL) with project number P16-25, which is (partly) financed by the Dutch Research Council (NWO).

References

[1] B. Lange, M. Itkina, and M. Kochenderfer, "Attention augmented ConvLSTM for environment prediction," *ArXiv*, type = Preprint, archivePrefix = arXiv, eprint = 2010.09662, Tech. Rep., 2020.

[2] E. A. Pool, J. F. Kooij, and D. M. Gavrila, "Crafted vs learned representations in predictive models—a case study on cyclist path prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 747–759, 2021.

[3] W. Luo, B. Yang, and R. F. a. Urtasun, "Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net," *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, pp. 3569–3577, 2018.

[4] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 116–121.

[5] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," *The International Journal of Robotics Research*, vol. 37, no. 8, pp. 841–866, 2018.

[6] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, and B. Caine, "Scalability in perception for autonomous driving: Waymo Open dataset," *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*, pp. 2446–2454, 2020.

[7] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[8] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *European Conference On Computer Vision*, pp. 694–711, 2016.

[9] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," *The Thirty-Seventh Asilomar Conference On Signals, Systems & Computers, 2003*, vol. 2, pp. 1398–1402, 2003.

[10] M. Itkina, K. Driggs-Campbell, and M. Kochenderfer, "Dynamic environment prediction in urban scenes using recurrent representation learning," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2052–2059, 2019.

[11] M. Toyungyernsub, M. Itkina, R. Senanayake, and M. Kochenderfer, "Double-Prong ConvLSTM for spatiotemporal occupancy prediction in dynamic environments," *ArXiv*, type = Preprint, archivePrefix = arXiv, eprint = 2011.09045, Tech. Rep., 2020.

[12] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," *2018 IEEE International Conference On Robotics And Automation (ICRA)*, pp. 2056–2063, 2018.

[13] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," *2019 International Conference On Robotics And Automation (ICRA)*, pp. 9299–9305, 2019.

[14] J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal Of Robotics Research*, vol. 37, pp. 492–512, 2018.

[15] N. Mohajerin and M. Rohani, "Multi-step prediction of occupancy grid maps with recurrent neural networks," *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*, pp. 10 600–10 608, 2019.

[16] Z. Wang, A. Bovik, H. Sheikh, and E. I. q. a. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions On Image Processing*, vol. 13, pp. 600–612, 2004.

[17] P. Wu, S. Chen, and D. M. Metaxas, "Joint perception and motion prediction for autonomous driving based on bird's eye view maps," *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*, pp. 11 385–11 395, 2020.

[18] S. Xingjian, 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*, 2015, pp. 802–810.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[20] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," *Advances in neural information processing systems*, vol. 29, pp. 64–72, 2016.

[21] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip, "PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5123–5132.

[22] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *ArXiv*, type = Preprint, archivePrefix = arXiv, eprint = 1605.08104, Tech. Rep., 2016.

[23] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73 – 101, 1964. [Online]. Available: https://doi.org/10.1214/aoms/1177703732

[24] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[25] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, "Kornia: an open source differentiable computer vision library for PyTorch," 10 2019.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[27] C. Tralie, "2D histogram Wasserstein distance via POT library," 2018. [Online]. Available: https://gist.github.com/ctralie/66352ae6ab06c009f02c705385a446f3

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv Preprint*, archivePrefix = arXiv, eprint = 1412.6980.

[29] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.