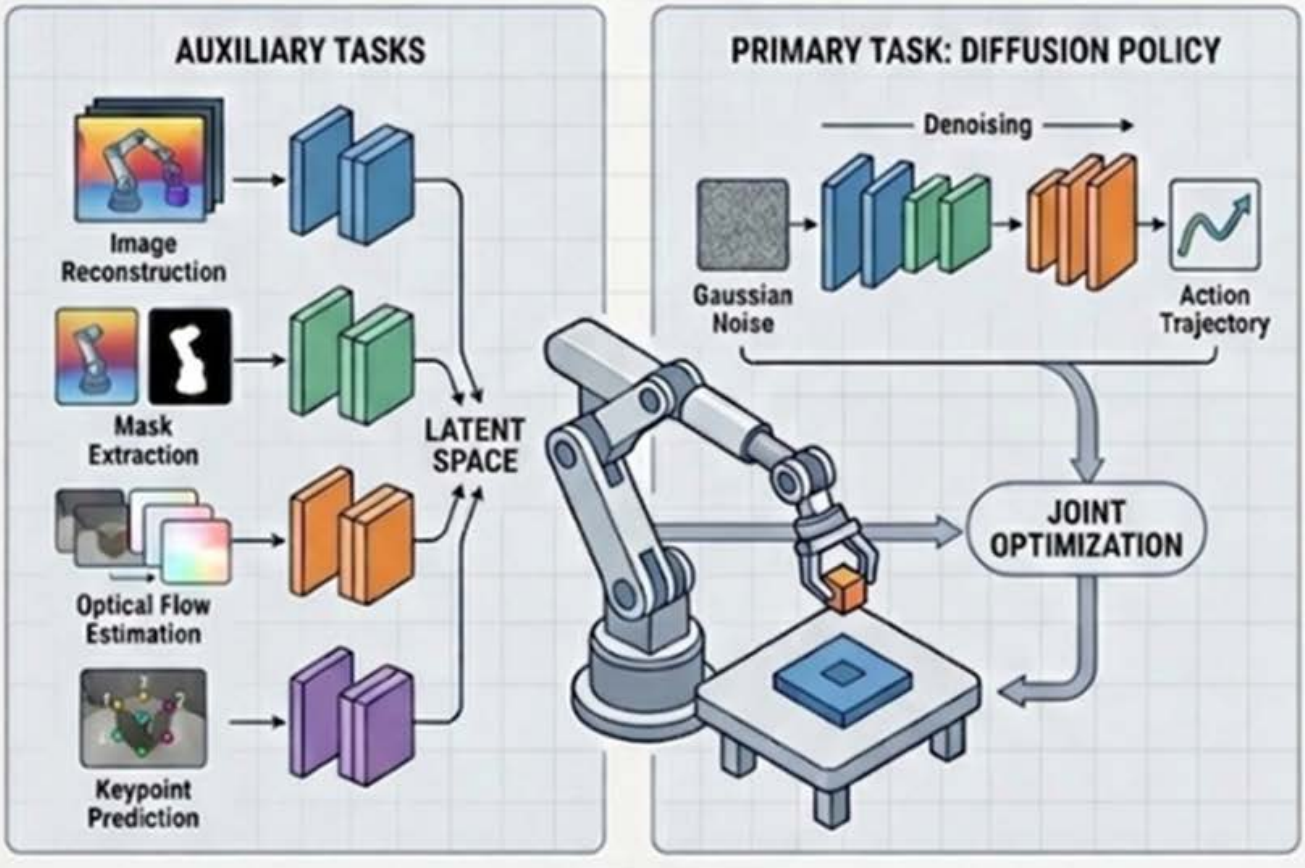
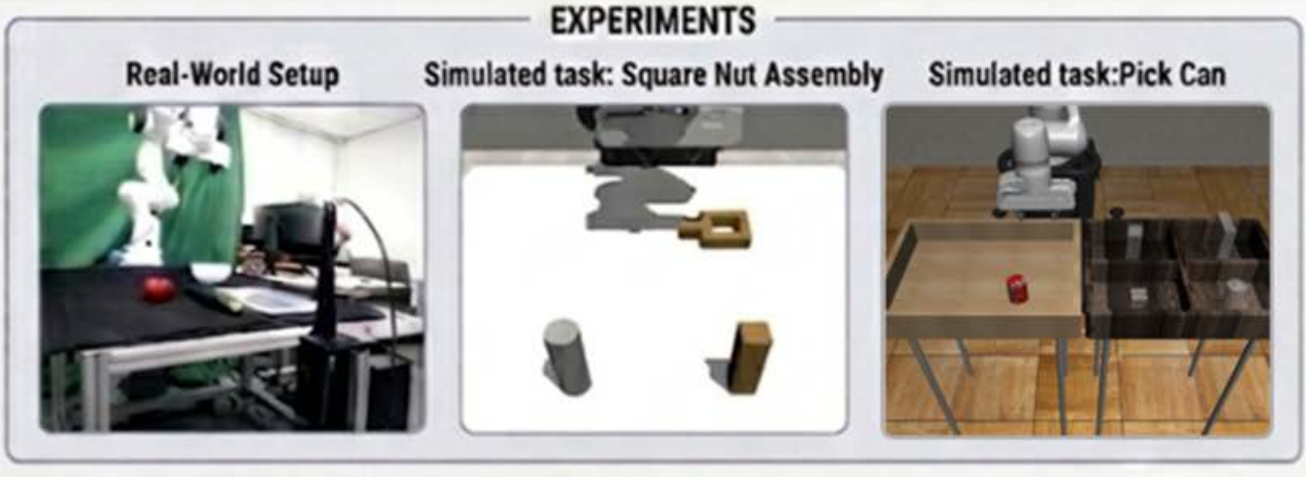
An Empirical Study on Auxiliary Task Joint-Training for Diffusion Policy

Qifan Luo





TU Delft

An Empirical Study on Auxiliary Task Joint-Training for Diffusion Policy

by

Qifan Luo

6227554

M.Sc. Robotics

Supervisor: Dr. Cosimo Della Santina
M.Sc. Zhaoting Li
Faculty: Cognitive Robotics, Mechanical Engineering, TU Delft

TU Delft

An Empirical Study on Auxiliary Task Joint-Training for Diffusion Policy

Qifan Luo (6227554)

***Abstract*— While Diffusion Policy has emerged as a powerful framework for robotic manipulation due to its expressiveness in modeling complex action distributions, its deployment is heavily constrained by high demonstration collection costs. This study presents a systematic empirical investigation into whether joint-training with visual auxiliary tasks can enhance the sample efficiency of diffusion policies under single-task spatial generalization (i.e., variations in object orientations and initial locations). Restricting observation inputs to raw 2D images and low-dimensional robot proprioception, we incorporate four candidate auxiliary tasks: image reconstruction, active object mask extraction, keypoint prediction, and optical flow estimation. We evaluate them with a joint-training framework across two simulated manipulation tasks and one real-world robotic task, using varying amounts of demonstration data. Our empirical findings demonstrate that joint-training with auxiliary tasks indeed provides sample efficiency benefits, particularly in intermediate data regimes. However, we observe that in certain cases, optimization conflicts and gradient interference between auxiliary and primary tasks diminish these benefits, especially in data-starved or data-rich regimes under simulated settings.**

I. Introduction

Robotic manipulation is the essential capacity of an autonomous agent to engage with its environment through physical actions like grasping, relocating, and assembling objects [1]. This field acts as the vital link between planning and execution, allowing machines to automate tasks that are either too dangerous or repetitive for humans. Consequently, it has become a cornerstone of modern manufacturing, logistics, and medical technology [2].

Traditionally, robotic systems relied on model-based control frameworks. These methods work well in structured settings like factory assembly lines. However, they often fail in unstructured, real-world environments. Their primary limitations include perceptual sensitivity and lack of adaptability. Variations in lighting or occlusions create sensory noise that rigid mathematical models cannot easily interpret. In addition, differences in object locations and poses introduce uncertainty that predefined models cannot handle effectively. As a result, these systems struggle to generalize across diverse scenarios and adapt to dynamic changes in the environment [3].

To overcome these hurdles, the field has transitioned toward data-driven strategies. By leveraging large-scale datasets and machine learning techniques, robotic systems have demonstrated an unprecedented capacity to perceive, adapt, and reason. This shift, supported by modern generative modeling, has enabled robots to achieve greater dexterity. Robots are now increasingly capable of managing unpredictable variables, such as changes in lighting, object placement, or occlusions. As a result, they can adapt to new situations and perform more complex tasks in dynamic, real-world environments [4].

Imitation Learning (IL) is a representative branch of data-driven strategies. It enables a robot to learn a policy by observing demonstrations from a human expert. The robot improves its performance by reducing the difference between its own actions and those of the expert. Among IL, Diffusion Policy (DP) [5] has become the mainstream because of its exceptional ability to model complex, multimodal action distributions and superior training stability. An overview of the structure for DP is shown in Figure 1a, in which the visual encoder is trained end-to-end. It consists of two main components: a visual encoder and a diffusion policy network. The visual encoder is intended to learn generalizable latent representations from high-dimensional sensory inputs (e.g., images) that preserve only the task-relevant information needed for policy learning. While the diffusion policy network is designed to generate the actual executable actions for the robotic hardware, conditioned on the latent representation.

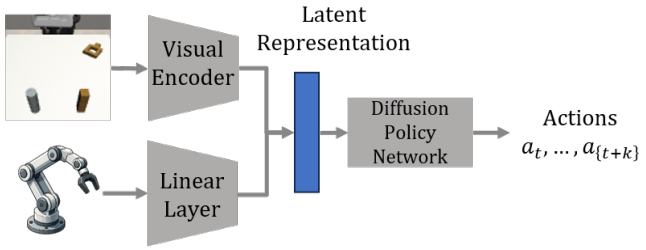
A major limitation of IL (including DP) is the heavy workload for human experts. Collecting demonstrations is time-consuming. One study shows that it takes 65 minutes to collect just 100 demonstrations [6]. Simple tasks need about 90 demonstrations, while complex or long-horizon tasks require more than 100 [5]. This leads to an important question: How can we improve sample efficiency? In other words, how can we reach a high success rate with fewer demonstrations?

Extensive research has been conducted to address this issue. This includes theoretical work that refines the sample complexity of offline methods [7], empirical studies on strategies for efficient data collection [8], and interactive algorithms that utilize novel forms of expert feedback [9]. Beyond policy improvement and data collection, another line of work focuses on pretraining visual encoders to reduce the dependence on expert demonstrations during fine-tuning [10], [11], [12], [13].

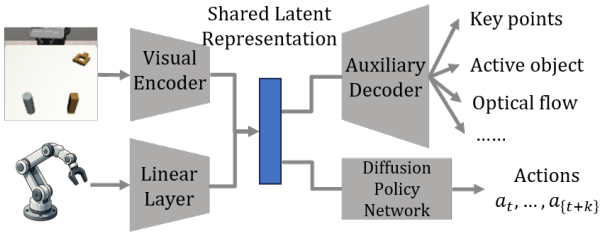
However, current research has overlooked the joint-training framework, which optimizes the visual encoder, auxiliary decoder, and imitation policy concurrently for targeted manipulation tasks [14]. This approach has already proven its effectiveness in some classic reinforcement learning tasks [15].

Therefore, our research question is: Can joint-training with auxiliary tasks provide sample efficiency benefits for diffusion policy under single-task spatial generalization (i.e., variations in object orientations and initial locations), compared to end-to-end training from scratch?

To address this research question, we integrate auxil-



(a) An overview of the structure for DP, in which the visual encoder is trained end-to-end.



(b) The architecture for a joint-training framework that integrates auxiliary visual representation learning into the diffusion policy training loop.

Fig. 1: Comparison between standard end-to-end Diffusion Policy and our joint-training framework.

iary visual representation learning into the diffusion policy training loop. By augmenting the standard objective with auxiliary tasks, we encourage the visual encoder to extract task-relevant features, thereby enhancing policy performance under limited data regimes. The architecture for this joint-training framework is illustrated in Figure 1b. In our implementation, we strictly constrain the sensory observation inputs to raw 2D images combined with standard robot proprioception. Through systematic experiments, we demonstrate that joint training with auxiliary tasks improves sample efficiency. This approach enables the diffusion policy to generalize effectively across variations in object pose while requiring fewer demonstrations than end-to-end training without using auxiliary tasks to guide the latent representation learning.

The contributions of this work are summarized as follows:

- Identify effective and compatible auxiliary tasks for the joint-training framework from a range of visual encoder training techniques.
- Establish reliable ground truth annotations for the identified auxiliary tasks.
- Develop task-specific decoders to accommodate the optimization objectives of different auxiliary tasks.
- Conduct extensive experiments under different demonstration quantities (i.e., 25%, 50%, 75%) to empirically verify whether competitive performance can be achieved with fewer demonstrations, in comparison with conventional end-to-end training from scratch.

The remainder of this report is organized as follows. Section II reviews the related work. Section III elaborates our implementation of joint training with auxiliary tasks. Sections IV and V present the results and analysis of our simulation and real-world experiments, respectively. Section VI concludes this work.

II. Related Works

A. Imitation Learning

1) Behavioral Cloning: Behavioral Cloning (BC) is a core method in robotic imitation learning. It learns visuomotor policies by performing supervised regression on human demonstrations. However, standard BC often suffers from the non-Markovian nature of human actions, complex multi-modal action distributions, and poor long-horizon generalization.

Mandlekar et al. established the BC-RNN as a standard baseline [16]. Through large-scale benchmarking across eight simulated and real-world manipulation tasks, they systematically showed that modeling temporal dependencies effectively handles the non-Markovian characteristics of human data and significantly outperforms standard BC. Similarly, Shafiullah et al. proposed the Behavior Transformer (BeT) [17], a transformer-based BC framework that captures long-range dependencies more robustly. A key limitation of both BC-RNN and BeT is that they require pre-specifying the number of modes in the action distribution, making it inherently difficult for them to fit the complex, multi-modal action patterns common in real-world manipulation.

To better model such complex action distributions, Florence et al. proposed an energy-based implicit BC method [18]. It can universally approximate complex, discontinuous, and set-valued distributions. This method achieved order-of-magnitude improvements in high-precision robotic tasks. However, it suffers from training instability due to its contrastive loss and has high inference latency caused by iterative optimization. Subsequently, Chi et al. introduced Diffusion Policy (DP) [5]. It solves the training instability problem via a stable denoising diffusion objective and consistently outperforms previous methods. In addition, by predicting action sequences with receding-horizon control, it achieves a favorable balance between smooth long-horizon planning and responsiveness to unexpected observations.

2) Interactive Imitation Learning: Interactive Imitation Learning (IIL) is a type of imitation learning that uses intermittent human feedback during robot execution to improve behavior online. The foundational work of modern IIL is the Dataset Aggregation (DAgger) framework proposed by Ross et al. [19]. To address the compounding error and covariate shift in offline BC, it reduces imitation learning to a no-regret online learning problem, iteratively aggregating expert demonstrations on the agent's on-policy states to align training and testing distributions. Nevertheless, it requires the expert to label every state visited by the agent, which demands too much effort from human supervisors and ignores useful information from situations where no intervention occurs.

To alleviate the heavy annotation burden of DAgger and fully exploit implicit feedback signals, Spencer et al. proposed Expert Intervention Learning (EIL) [20]. It formalizes both explicit intervention and implicit non-intervention feedback as constraints on the agent's value function, and trains

the policy via efficient no-regret online learning. However, it still follows the point-label assumption of BC—where each state is assumed to have one optimal action label—which makes it sensitive to overfitting when human feedback is noisy.

To overcome the overfitting issue caused by the point-label assumption, Li et al. proposed Contrastive policy Learning from Interactive Corrections (CLIC) [21]. It constructs desired action spaces from human corrections instead of relying on single optimal action labels, and designs a KL-divergence loss to stably train energy-based policies. However, this approach is incapable of handling high-dimensional inputs, such as images. To overcome this limitation, Li et al. subsequently introduced the set-supervised diffusion policy (SDP) [22]. By integrating the desired action space with DP, SDP enables learning from both positive and negative samples, thereby achieving state-of-the-art performance.

B. Sample Efficiency in Imitation Learning

Current research concerning sample efficiency in imitation learning primarily focuses on theoretical analysis and data collection strategies. Foster et al. [7] provide a theoretical reanalysis of BC, showing that log-loss BC achieves horizon-independent sample complexity for deterministic experts, closing the perceived gap with online methods. Li and Zhang [9] reexamine the sample efficiency of imitation learning by shifting the focus from trajectory-level to state-wise annotation costs. Their work demonstrates that measuring annotation cost per state (rather than per trajectory) reveals significant advantages for IIL, especially when combining offline demonstrations with targeted online queries.

Regarding data collection strategies, Lin et al. [8] empirically investigate data scaling laws for robotic manipulation, revealing that policy generalization follows a power-law relationship with the number of training environments and objects. Critically, they demonstrate that environmental and object diversity far outweighs raw number of demonstrations.

C. Visual Encoder Pretraining

Visual encoder pretraining aims to learn general-purpose latent representations through large-scale pre-training, typically using self-supervised or contrastive objectives to distill task-relevant features. These pretrained encoders serve as robust feature extraction backbones, providing strong visual representation priors for diverse downstream manipulation tasks, thereby enhancing the efficiency and performance of policy learning. An overview of the pre-training and fine-tuning frameworks is presented in Figure 2.

R3M [23] pre-trains a universal visual representation for robotic manipulation on Ego4D [24], combining time-contrastive learning, video-language alignment, and L1 sparsity penalty. It achieves a superior few-shot imitation performance on both simulated and real robots. MVP [12] adopts the masked autoencoder for self-supervised pre-training on Ego4D and other egocentric video datasets. It validates the effectiveness of masked visual pre-training on real-world robotic tasks, and realizes synchronous scaling of model

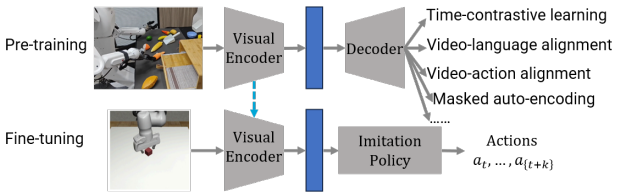


Fig. 2: Overview of the pre-training and fine-tuning framework.

and data size. VC-1 [13] constructs a general artificial visual cortex for embodied intelligence by pre-training Vision Transformers on a massive, diverse mixture of egocentric human video datasets using a masked auto-encoding paradigm. It verifies strong generalization across diverse robotic tasks, including navigation, manipulation, and locomotion in both simulation and the real world. HRP [10] automatically extracts three types of human affordance labels (contact points, hand poses, active objects) from Ego4D videos. It fine-tunes existing pre-trained models with these labels, significantly improving the robustness of imitation learning across camera views and robot morphologies. MCR [11] first introduces the metric of manipulation centricity and validates its strong positive correlation with downstream robotic manipulation performance, establishing a new evaluation paradigm for robotic visual representations. Pre-trained on the large-scale DROID robot dataset [25], it learns manipulation-centric latent representations via a joint objective of dynamics alignment, action prediction, and temporal contrastive learning. It achieves substantial performance gains in few-shot imitation learning across both simulation and real-world robotic manipulation tasks.

Visual encoder pretraining enables the encoding of environmental dynamics, physical interaction patterns, and temporal causal dependencies via carefully designed contrastive learning or self-supervised training objectives. This paradigm, in turn, supports few-shot generalization to downstream tasks. However, this paradigm still faces non-negligible inherent limitations. For specific downstream manipulation tasks completely unseen in the pre-training corpus, it can only achieve a modest success rate of no more than 80% even with 30-50 fine-tuning. This performance bottleneck is largely constrained by cross-task dynamics, distribution shifts, and the inherent mismatch of manipulation scenarios between pre-training and target tasks.

III. Method

Building upon the related works, this section will elaborate on the auxiliary tasks selection, ground truth establishment, and task-specific decoders in detail.

A. Auxiliary Tasks Selection

To select auxiliary tasks that are both effective and compatible with the joint-training framework, we base our selection on a comprehensive survey [26], which reviews 120 research articles within the section *Learning with Auxiliary Tasks*.

Figure 3 presents a mindmap that categorizes various auxiliary tasks. Through a rigorous screening process, we exclude ineffective or incompatible tasks (highlighted in red) and select four promising candidate tasks for subsequent empirical evaluation (highlighted in green). In the following subsections, we provide a detailed analysis of each task to justify its suitability or limitations within our framework.

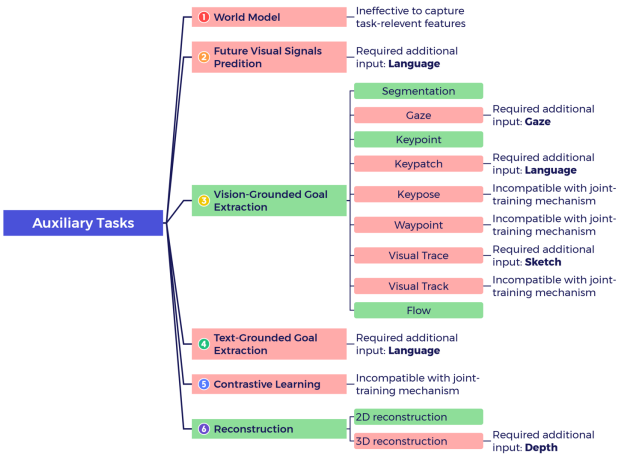


Fig. 3: An overview of various auxiliary tasks.

1) World Model: A world model learns an internal representation of environment dynamics. Observation at the next time (o_{t+1}) step is predicted based on observation and action at the current time step (o_t, a_t). World models focus on every pixel change in the environment. This forces the visual encoder to waste resources on irrelevant details, like background variations, making it hard to capture task-relevant features.

2) Future Visual Signals Prediction: In contrast to world models, these methods synthesize future frames without learning explicit action-conditioned dynamics. However, it relies on natural language input to guide predictions. This requirement for multi-modal context deviates from our vision-only formulation, making them unsuited for our joint-training framework.

3) Vision-Grounded Goal Extraction: These methods rely on visual signals to extract task-relevant features. However, certain approaches necessitate supplementary modalities—such as gaze, natural language, or sketches—to identify the most informative regions within raw observations. Furthermore, tasks focusing on keyposes and waypoints utilize $SE(3)$ representations as precise geometric targets for control, introducing a supervision format that is structurally misaligned with standard 2D image joint training. Finally, visual tracking methods encode trajectories of scene points across time, employing them as an action or guidance space to bridge perception and control. While effective, this paradigm demands highly complex architectures, thereby deviating from the primary motivation of utilizing lightweight models for auxiliary tasks during joint training.

In contrast, tasks such as semantic segmentation, keypoint detection, and optical flow estimation emerge as both highly effective and structurally compatible choices.

4) Text-Grounded Goal Extraction: Recent advances in large language models have facilitated the rise of natural language-conditioned goal extraction methods. They link linguistic references directly to specific entities in the environment, thereby reducing ambiguity and alleviating cognitive load. Nevertheless, because our framework restricts its inputs strictly to 2D images, incorporating such language-conditioned tasks is unfeasible due to the absence of textual inputs.

5) Contrastive Learning: Contrastive learning trains models to extract meaningful representations by pulling embedding vectors of similar pairs closer together while pushing dissimilar pairs apart. Common objectives include time-contrastive alignment to advance sequential states toward a goal, language-conditioned alignment to map observations to textual instructions, and video-to-action alignment to link visual frames with execution behaviors. However, the reliance on pairwise loss functions introduces optimization conflicts during joint training.

6) Reconstruction: The central idea of these methods is to recover raw observations to infer the underlying structure of the environment, thereby promoting the learning of rich, structured representations. While 2D reconstruction seamlessly integrates with joint training, 3D reconstruction relies on depth-dependent data, such as point clouds or multi-view images. This dependency goes beyond the scope of our standard 2D input framework.

Eventually, four auxiliary tasks are selected, as detailed in Table I.

TABLE I: Auxiliary tasks selection and justifications

Task	Category	Justification
Image Reconstruction	Object-centric	Capture rich information including relationships between manipulator, environment, and objects
Keypoints Prediction	Object-centric	Capture the state-transition information, including the grasp and release points
Mask Extraction	Object-centric	Capture the object-centric information, including the manipulated object and target
Optical Flow Estimation	Motion-centric	Capture the motion of pixels in the image plane

B. Ground Truth Establishment

Based on the previous discussion, image reconstruction, active object mask extraction, optical flow estimation, and keypoints prediction are selected as auxiliary tasks, as they are regarded as both effective and compatible for joint training with the diffusion policy. Each task provides unique supervisory guidance, encouraging the visual encoder to extract meaningful representations from raw observations that facilitate policy learning. To train these auxiliary objectives, reliable ground truth must be established to supervise the learning process.

1) Image: Typically, image frames are represented as tensors with a shape of $C \times H \times W$ (Channels $\times$ Height $\times$ Width), where $C = 3$ for RGB data. The pixel values are usually normalized to a standard range, such as $[-1, 1]$.

Image reconstruction is formulated as a self-supervised auxiliary task. In this setting, the model is trained to reconstruct the original input observation, which itself serves as the ground-truth supervisory signal.

2) Active Object Mask: Active objects are defined as the objects being actively manipulated by the robot and the corresponding targets involved in the task.

The active object masks are represented as binary tensors. Each individual frame mask has a shape of $H \times W$, representing a single-channel spatial grid where H and W denote the frame height and width, respectively. The pixel values are strictly binary and normalized to the range $\{0, 1\}$. A pixel value of 1 signifies that the pixel belongs to an active object (manipulated object or target), while a pixel value of 0 denotes the background, the robot structure, or other irrelevant entities.

In the simulation environment, the ground truth for active object masks is accessible through structured semantic segmentation. This process leverages rendered object-ID buffers, in which each object instance is assigned a unique integer identifier during rendering. The complete procedure is described in Algorithm 1.

Algorithm 1: Active Object Mask Ground Truth Establishment in Simulation

Input: Trajectory buffer $\mathcal{B}$, Environment configuration $\mathcal{C}$, Active object IDs $\mathcal{I}_{\text{active}}$
Output: Compressed mask files for each episode
Initialize simulation environment Env via $\mathcal{C}$;
for $i = 0$ **to** $N - 1$ **do**
  ▷ Load saved states from the dataset
  $\tau \leftarrow \text{LoadTrajectory}(\mathcal{B}, i)$
  $\mathcal{M}_{\text{agt\_view}} \leftarrow [\,]$, $\mathcal{M}_{\text{hand\_view}} \leftarrow [\,]$
  for $t = 0$ **to** $|\tau| - 1$ **do**
    $\mathcal{O}_t \leftarrow \text{ResetEnvironmentToState}(\text{Env}, \tau, t)$
    ▷ Get raw segmentation maps
    $S_{\text{agt\_view}} \leftarrow \mathcal{O}_t$["agt_view_seg"]
    $S_{\text{hand\_view}} \leftarrow \mathcal{O}_t$["hand_view_seg"]
    ▷ Filter by active object IDs (shape: $H \times W$)
    $M_{\text{agt\_view}} \leftarrow \mathbb{I}(S_{\text{agt\_view}} \in \mathcal{I}_{\text{active}})$
    $M_{\text{hand\_view}} \leftarrow \mathbb{I}(S_{\text{hand\_view}} \in \mathcal{I}_{\text{active}})$
    Append $M_{\text{agt\_view}}$ to $\mathcal{M}_{\text{agt\_view}}$
    Append $M_{\text{hand\_view}}$ to $\mathcal{M}_{\text{hand\_view}}$
  ▷ Shape: $T \times H \times W$
  SaveMasks$(i, \mathcal{M}_{\text{agt\_view}}, \mathcal{M}_{\text{hand\_view}})$

In real-world scenarios, establishing ground truth for active object masks can be achieved by leveraging pre-trained vision foundation models, such as Segment Anything (SAM) [27]. To streamline this process, we employ RoboIntern-Tool [28], a comprehensive and semi-automated annotation pipeline. By integrating SAM2 with a lightweight GUI, this tool facilitates efficient large-scale demonstration labeling. Furthermore, it supports iterative quality control through intuitive visualization and multi-round verification, ensuring the precision of the generated masks. Figure 4 presents a flow chart of this pipeline.

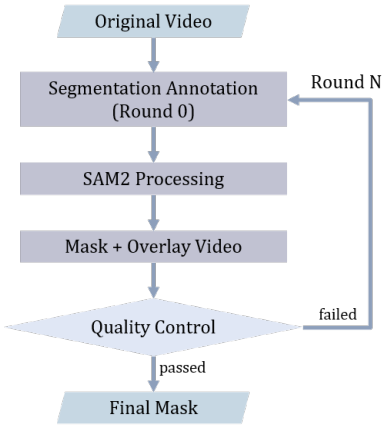


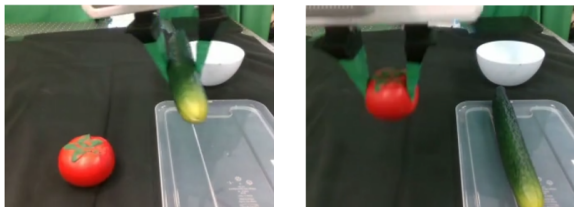
Fig. 4: The workflow for RoboIntern-Tool.

Figure 5 presents an example of masks ground truth in the simulation, corresponding to a task where the manipulator grasps a square nut and assembles it onto a cubic peg.

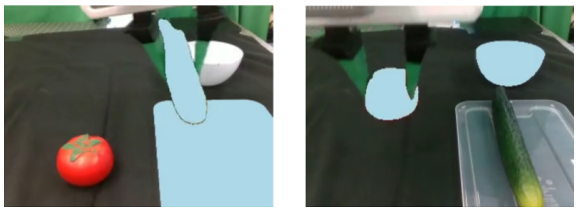
(a) Raw observation (b) Masks

Fig. 5: An example of active object masks in simulation.

Figure 6 illustrates the ground truth of the annotated active object masks in the real world. Notably, when multiple objects are present, we mask only the currently manipulated object and its corresponding target.



(a) Raw observation from the fixed workspace camera



(b) Corresponding masks

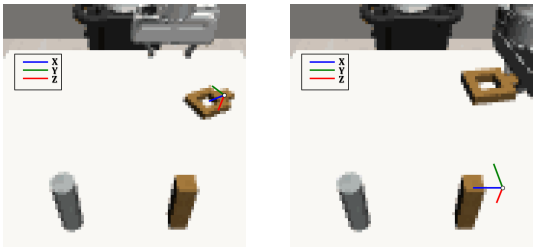
Fig. 6: Examples of the annotated active object masks in the real world.

3) Keypoint: Keypoints are defined as the critical transition coordinates within a manipulation trajectory, specifically capturing the grasp point and the release point where the state of interaction with an object undergoes a fundamental change.

The keypoints are represented as a tensor with a designated shape of 4×2. This matrix encapsulates both the two-dimensional pixel coordinates (u, v) on the image plane and the three-dimensional Cartesian positions (X, Y, Z) representing the spatial references.

The ground truth for keypoints is precisely determined by the displacement variations of the robotic gripper. Specifically, a keypoint is identified at the exact timestamp when a transitional variation in the gripper's displacement occurs, provided that its state remains stable and unchanged for a subsequent period. Once these keypoint timestamps are captured, the ground truth coordinates are generated through sequential geometric projections. First, the 3D gripper position defined in the robot base frame is transformed into the camera coordinate system using the extrinsic coordinate transformation matrix. Subsequently, these spatial coordinates are projected onto the 2D image plane via the camera intrinsic matrix. The comprehensive procedure is formalized in Algorithm 2.

An example of the keypoints ground truth is illustrated in Figure 7.



(a) Grasp point (b) Release point

Fig. 7: An example of the extracted active object masks.

4) Optical Flow: Optical flow is defined as the apparent motion of individual pixels across the image plane between consecutive video frames. It provides rich, dense, and instantaneous dynamic information regarding the temporal evolution of the visual environment. For robot policy learning, incorporating optical flow offers critical advantages: it explicitly captures the underlying velocity fields of both moving objects and the robot's own components, thereby serving as a powerful predictive cue that guides the network to reason about temporal dynamics, causality, and future states without requiring long history stacks.

The extracted optical flow fields are represented as continuous floating-point tensors. Each flow field map has a shape of $H \times W \times 2$, where H and W denote the frame height and width, respectively, matching the structural dimensions of the spatial grid. The final channel dimension of size 2 corresponds to the horizontal and vertical displacement vectors, $(\Delta u, \Delta v)$, for each individual pixel. The numerical values of the flow vectors are real numbers within a continuous range ($\Delta u \in [-W, W], \Delta v \in [-H, H]$), representing the exact sub-pixel coordinate offsets from the source frame to the target frame.

Algorithm 2: Keypoints Ground Truth Establishment

Input: Environment configuration $\mathcal{C}$, Threshold ϵ, Trajectory buffer $\mathcal{B}$, Intrinsic matrix $\mathbf{K}$, Extrinsic matrix $\mathbf{T}_{\text{cam}\leftarrow\text{world}}$
Output: Compressed keypoint files for each episode
Initialize simulation environment Env via $\mathcal{C}$
for $i = 0$ **to** $N - 1$ **do**
  ▷ Load saved trajectory from the dataset
  $\tau \leftarrow \text{LoadTrajectory}(\mathcal{B}, i)$
  $\mathcal{D} \leftarrow [\,]$ ▷Initialize gripper data state list
  for $t = 0$ **to** $|\tau| - 1$ **do**
    $\mathcal{O}_t \leftarrow \text{ResetEnvironmentToState}(\text{Env}, \tau, t)$
    ▷ Extract gripper state
    $q_1, q_2 \leftarrow \text{ExtractGripperPosition}(\mathcal{O}_t)$
    Append (q_1, q_2, t) to $\mathcal{D}$
  ▷ Identify transition indices
  $\mathcal{K}_{\text{indices}} \leftarrow [\,]$
  PositionChanged $\leftarrow$ False
  for $j = 1$ **to** $|\mathcal{D}| - 1$ **do**
    $(q_1^{j-1}, q_2^{j-1}, \_) \leftarrow \mathcal{D}[j-1]$
    $(q_1^{j}, q_2^{j}, t_j) \leftarrow \mathcal{D}[j]$
    if $(|q_1^j - q_1^{j-1}| > \epsilon)$ ***or*** $(|q_2^j - q_2^{j-1}| > \epsilon)$ **then**
      PositionChanged $\leftarrow$ True
    if *PositionChanged* ***and*** $j < |\mathcal{D}| - 1$ **then**
      $(q_1^{j+1}, q_2^{j+1}, \_) \leftarrow \mathcal{D}[j+1]$
      if $(|q_1^j - q_1^{j+1}| < \epsilon)$ ***and*** $(|q_2^j - q_2^{j+1}| < \epsilon)$ **then**
        Append t_j to $\mathcal{K}_{\text{indices}}$
        PositionChanged $\leftarrow$ False ▷Reset
  ▷ Project spatial 3D keypoints onto image plane
  $P \leftarrow [\,]$
  for *each keypoint index* $k \in \mathcal{K}_{indices}$ **do**
    $\mathcal{O}_k \leftarrow \text{ResetEnvironmentToState}(\text{Env}, \tau, k)$
    ▷ 3D world coordinates of End-Effector (eef)
    $\mathbf{x}_{\text{world}} \leftarrow [X, Y, Z]^T \leftarrow \mathcal{O}_k$["eef_pos]
    ▷ 2.World $\rightarrow$ Camera
    $\mathbf{x}_{\text{cam}} \leftarrow \mathbf{T}_{\text{cam}\leftarrow\text{world}} \cdot [\mathbf{x}_{\text{world}}, 1]^T$
    ▷ 3.Camera $\rightarrow$ Image plane
    $z_{\text{cam}} \cdot [u, v, 1]^T \leftarrow \mathbf{K} \cdot \mathbf{x}_{\text{cam}}$
    $\mathbf{p}_{\text{img}} \leftarrow [u, v]^T$
    Append $\mathbf{p}_{\text{img}}$ to P
  SaveKeypoints(i, P) ▷Shape: $|\mathcal{K}_{\text{indices}}| \times 4 \times 2$

The ground truth for the optical flow fields is established leveraging a global matching framework and a differentiable matching formulation [29]. Given two consecutive rendered feature maps $F_1, F_2 \in \mathbb{R}^{H \times W \times D}$, the dense global correlation matrix $C \in \mathbb{R}^{H \times W \times H \times W}$ is computed efficiently via matrix multiplication:

$$C = \frac{F_1 F_2^T}{\sqrt{D}} \tag{1}$$

To ensure end-to-end differentiability and achieve sub-pixel level accuracy, a soft-argmax operation is performed by normalizing the last two dimensions of C with a softmax function, yielding the matching distribution $M \in \mathbb{R}^{H\times W\times H\times W}$.

$$M = \text{softmax}(C) \tag{2}$$

The predicted coordinates $\hat{G} \in \mathbb{R}^{H\times W\times 2}$ are then computed as the weighted average of the standard 2D coordinate grid $G \in \mathbb{R}^{H\times W\times 2}$ parameterized by M:

$$\hat{G} = MG \tag{3}$$

Finally, the precise ground truth optical flow tensor V is directly derived by calculating the explicit coordinate difference between the matched correspondence and the original pixel grid:

$$V = \hat{G} - G \in \mathbb{R}^{H\times W\times 2} \tag{4}$$

This formulation ensures that the generated ground truth preserves continuous displacements, effectively resolving the long-standing challenge of large displacements while maintaining pixel-level structural consistency. Figure 8 is an example that depicts this process for clarity.

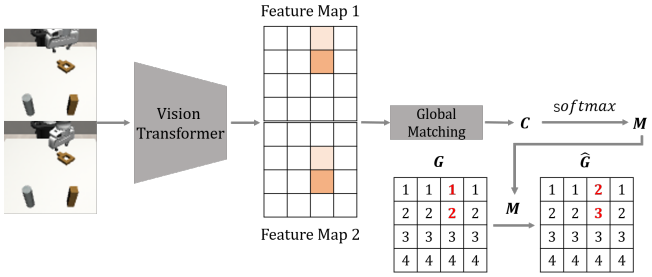
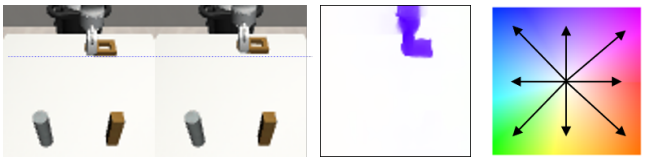


Fig. 8: Learning optical flow from global matching.

An example of the ground truth for the optical flow is shown in Figure 9. The two images on the left show the raw observations at time steps t and $t+1$. The middle image presents the corresponding ground-truth optical flow field derived from these observations. The rightmost image provides the color-coding legend, which maps hue to the direction of motion. In this instance, the manipulator is lifting a square nut, resulting in a predominantly purple flow field that indicates upward movement.



(a) Raw observations from two successive frames (b) Derived optical flow (c) Color-coding legend

Fig. 9: An example of ground truth for optical flow.

C. Task-Specific Decoders Design

As detailed in the preceding section, the data dimensions vary across the four auxiliary tasks; a comprehensive summary of their respective values and structural shapes is provided in Table II.

TABLE II: Summary of data types, value ranges, and shapes.

Data	Type & Value Range	Shape
Image	float $\in [-1, 1]$	$H \times W \times 3$
Mask	int $\in \{0, 1\}$	$H \times W$
Keypoints	int $\in [-W, W] \times [-H, H]$	4×2
Optical Flow	float $\in [-W, W] \times [-H, H]$	$H \times W \times 2$

While the target outputs for image reconstruction, mask extraction, and optical flow estimation all share a spatial resolution grid related to $H \times W$ (differing only in their channel dimensions and value ranges), the keypoint prediction task yields a dense yet distinct structure of shape 4×2. To accommodate these structural and semantic discrepancies, we present two tailored decoder architectures specifically optimized for their respective data representations.

1) CNN-Based Decoder for Spatial Map Prediction: For pixel-level dense prediction tasks (i.e., image reconstruction, mask extraction, and optical flow estimation), the decoder is designed based on Deep Convolutional Generative Adversarial Networks (DCGAN) [30]. This architecture is highly efficient in upsampling dense structural maps from a low-dimensional latent bottleneck while maintaining superior generative capabilities.

Specifically, given a 256-dimensional latent representation, it is first projected and reshaped into a low-resolution feature map of shape $H' \times W' \times 512$. Specifically, H' and W' represent the spatial resolution of the encoder's output, derived by passing the raw observations through a ResNet-18 backbone that performs a $32\times$ downsampling, inclusive of padding. A sequence of five consecutive convolutional and upsampling layers (CONV 1 to CONV 5) is then applied, progressively doubling the spatial resolution while halving the channel capacity ($512 \to 256 \to 128 \to 64 \to 32 \to 16$).

To produce the task-specific outputs, three independent convolutional heads (CONV 6) are appended to decode the shared features:

- **Image Reconstruction:** Outputs a feature map of shape $[32H' \times 32W' \times 3]$, which is further upsampled via bilinear interpolation to match the original image dimensions $[H \times W \times 3]$.
- **Optical Flow Estimation:** Generates a $[32H' \times 32W' \times 2]$ dense flow field, followed by interpolation to the target shape $[H \times W \times 2]$.
- **Mask Extraction:** Outputs a $[32H' \times 32W' \times 1]$ logit map, passes through a Sigmoid activation to constrain values within $[0, 1]$, and is interpolated to the final shape $[H \times W]$.

Figure 10 illustrates the architecture of the designed CNN-based decoder in detail.

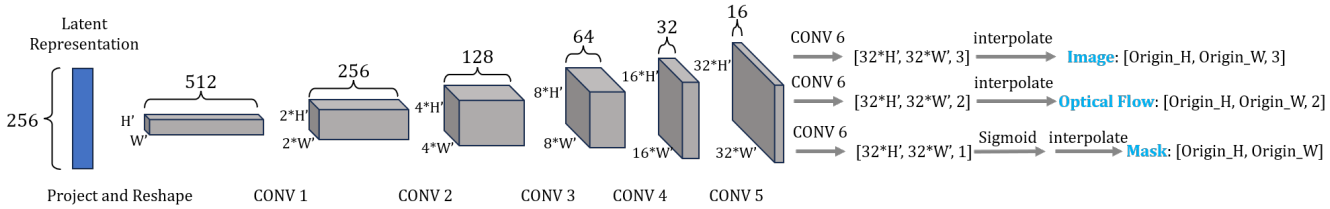


Fig. 10: Architecture for CNN-based decoder for spatial map prediction.

2) MLP-Based Decoder for Keypoints Regression: In contrast to the dense spatial maps, keypoint prediction requires mapping the global context to a compact coordinate set. Therefore, we implement a lightweight regression decoder [10], [31], utilizing a Multi-Layer Perceptron (MLP) network.

The architecture takes the identical 256-dimensional latent representation as input and progressively condenses the feature space through two fully connected hidden layers with 128 and 64 units, respectively:

$$h_1 = \sigma(W_1 x + b_1), \quad h_2 = \sigma(W_2 h_1 + b_2) \tag{5}$$

where $\sigma(\cdot)$ denotes the ReLU activation function, and $x \in \mathbb{R}^{256}$ represents the bottleneck vector.

The final output layer projects the 64-dimensional feature vector into a 4×2 matrix, directly regressing the continuous coordinates for the 4 target keypoints. This structural design significantly reduces computational overhead while maintaining high precision for low-dimensional coordinate regression.

Figure 11 illustrates the architecture of the designed MLP-based decoder in detail.

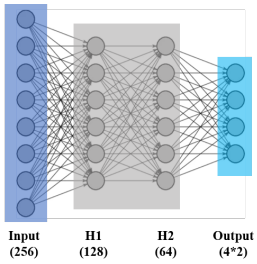


Fig. 11: Architecture for MLP-based decoder for compact keypoints prediction.

D. Implementation

The implementation details are described in Figure 12.

To map the learned visual representations to precise robotic execution, we adopt the Set-Supervised Diffusion Policy (SDP) [22] as our primary policy network. As a recent state-of-the-art enhancement of the standard DP, SDP substantially improves the generation quality by incorporating human corrections.

Crucially, the parameters of the Visual Encoder are updated via a joint objective function that balances the task-specific behavioral cloning and the representation-enriching auxiliary objectives. The total loss function is formulated as a weighted summation:

$$\mathcal{L}_{\text{total}} = w \cdot \mathcal{L}_{\text{auxiliary}} + \mathcal{L}_{\text{action}} \tag{6}$$

where $\mathcal{L}_{\text{action}}$ represents the action loss computed against expert demonstrations, and $\mathcal{L}_{\text{auxiliary}}$ introduces additional supervisory guidance for visual feature learning. The hyper-parameter w serves as a balancing weight to ensure that the auxiliary gradients enrich the latent representation without compromising the primary policy learning.

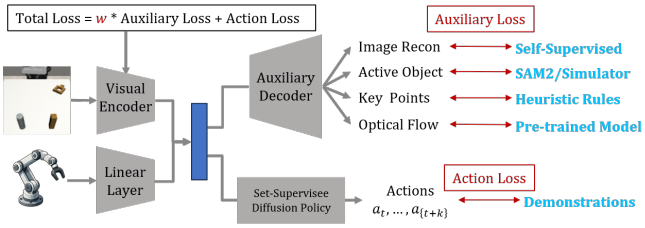


Fig. 12: Implementation details.

Among the auxiliary tasks, optical flow estimation is specifically tailored. Instead of processing a single static frame, two consecutive raw observations are taken as input. The objective is to estimate the optical flow at the first frame. Providing this explicit temporal context guides the visual encoder to inherently capture the dynamic transitions of the environment. Consequently, it forces the model to focus on moving pixels within the image plane, filtering out static background noise.

Table III outlines the specific implementation details, including the inputs, outputs, and loss types for each task.

TABLE III: Implementation details for each auxiliary task.

Auxiliary Task	**Input**	**Output**	**Loss Type**
Image Reconstruction	o_t	o_t	L2
Mask Extraction	o_t	Mask_t	Binary Cross-Entropy
Keypoints Prediction	o_t	Keypoints_t	L2
Optical Flow Estimation	o_t, o_{t+1}	Flow_t	L2

IV. Simulation Experiments

This section evaluates joint-training with auxiliary tasks in simulation. First, the experimental configurations are outlined. Subsequently, we analyze the performance of four auxiliary tasks and investigate how varying their loss weights impacts the balance between task semantics and visual noise.

A. Training Setup

1) Environment: To fully explore the influence of joint-training with auxiliary tasks, we first conduct extensive experiments in simulation. We adopt Robosuite [32] as our simulation environment because it provides high-fidelity physics simulation and a standardized suite of manipulation benchmarks. Furthermore, it offers seamless access to instance segmentation masks and precise camera intrinsic and transformation matrices. This allows us to effortlessly extract rich semantic and geometric information—such as object masks and keypoints—to facilitate the comprehensive evaluation of various auxiliary tasks.

2) Settings: Specifically, we benchmark various auxiliary tasks across three distinct experimental settings derived from two representative manipulation tasks: *Square Nut Assembly* and *Pick Can*. The Square Nut Assembly task is a high-precision insertion problem requiring meticulous grasping and alignment. To investigate the impact of visual complexity, we evaluate this task under two distinct configurations: one with a pristine, clear background that minimizes visual disturbances, and another augmented with realistic shadow rendering to introduce subtle visual depth cues and lighting variations. Conversely, the Pick Can task features a more structured pick-and-place pipeline but introduces a complementary challenge: a cluttered workspace with severe visual disturbances designed to rigorously test the robustness of the robot's perception system. Figure 13 illustrates these three settings derived from two representative manipulation tasks.

(a) Square (b) Square-shadow (c) PickCan-shadow

Fig. 13: Three settings for simulation experiments.

3) Auxiliary Loss Weight: The total optimization objective is formulated as the weighted summation of the auxiliary loss and the primary action loss, as detailed in Equation 6. Consequently, the weight w serves as a critical hyperparameter governing multi-task balance. We initialize this parameter at a conservative value ($w = 0.2$) to ensure that the auxiliary objectives do not overwhelm the gradients of the primary action loss during the early stages of training. In cases where the auxiliary representations fail to converge effectively, or patterns remain poorly recognized, we marginally increase the weight to $w = 0.5$. This adaptive, step-wise tuning strategy ensures that the auxiliary tasks successfully shape the latent representation without compromising the convergence or performance of the primary policy.

4) Demonstration Quantities: To systematically evaluate the sample efficiency benefits provided by joint-training with auxiliary tasks, we conduct experiments using varying amounts of expert demonstrations. For each manipulation task, we collect a total of 200 high-quality expert trajectories. We then derive four experimental subsets by sampling the first 25%, 50%, 75%, and 100% of these trajectories, respectively.

B. Results

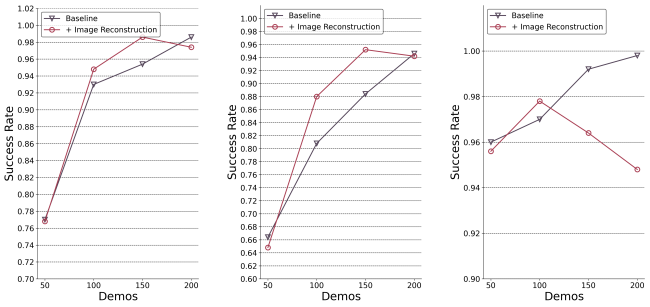
Table IV provides the simulation results for both the baseline and joint training with four auxiliary tasks, averaged across three experimental settings. Full results are detailed in Tables VII, VIII, and IX of the Appendix. It is worth noting that each result is averaged over 2 training seeds. Each seed reports the evaluation results averaged over the last 10 checkpoints. Each checkpoint is evaluated across 25 scenarios.

TABLE IV: Success rates averaged across three experimental settings (+*X* indicates joint training with *X*).

Demo Quantity	50	100	150	200
Baseline	0.798	0.903	0.943	0.977
+ Image Reconstruction	0.791	**0.934**	**0.967**	0.955
+ Mask Extraction	0.784	**0.936**	**0.958**	0.972
+ Optical Flow Extimation	0.799	0.856	0.896	0.939
+ Keypoint Prediction	0.644	0.809	0.914	0.897

In the following analysis, we report the best performance achieved across different auxiliary loss weights.

1) Image Reconstruction: Results of joint training with image reconstruction under three settings are presented in Figure 14.



(a) Square (b) Square-shadow (c) PickCan-shadow

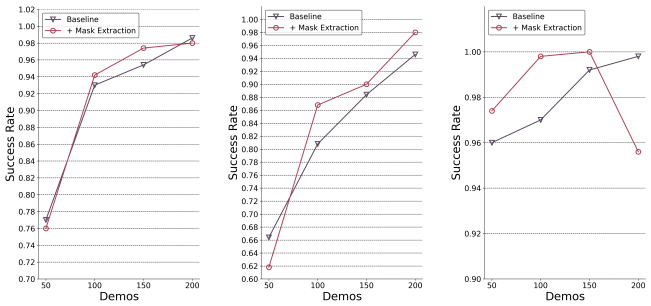
Fig. 14: Results for joint training with image reconstruction.

Several key trends can be observed from empirical results:

- **Sample Efficiency Benefits in the intermediate Regime**: In the intermediate demonstration regime (i.e., 100 and/or 150 demonstrations), joint training with image reconstruction outperforms the baseline, with the performance curves remaining at the top-left of the baseline. This trend indicates a substantial improvement in sample efficiency.

- **Performance Trade-offs in Low-Data Settings**: Conversely, when the number of demonstrations is highly limited (i.e., 50), joint training with image reconstruction marginally underperforms compared to the baseline across all configurations. In extremely low-data regimes, introducing an auxiliary task alongside the primary policy learning objective creates competing gradients, which ultimately distract the network and hinder its ability to converge on an effective control strategy.
- **Provide almost no benefits in Data-Rich Region**: As data becomes abundant, joint training with image reconstruction no longer outperforms the baseline (e.g., beyond 150 in Figures 14a and 14b, and beyond 100 in Figure 14c). While pixel-level reconstruction is highly beneficial for establishing basic semantic grounding early on, it fundamentally optimizes for visual fidelity rather than task-relevant semantics. With abundant data, the baseline policy can natively infer robust features directly from the behavior cloning loss.

2) Active Object Mask Extraction: Results of joint training with active object mask extraction under three settings are presented in Figure 15.



(a) Square (b) Square-shadow (c) PickCan-shadow

Fig. 15: Results for joint training with active object mask extraction.

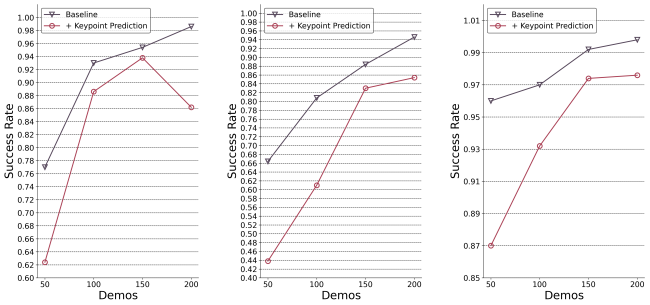
Similar trends are observed when joint training with active object mask extraction in data-scarce and mid-data regimes. In both configurations, the auxiliary task fails to outperform the baseline in data-scarce regimes due to optimization bottlenecks, yet provides clear sample efficiency benefits in mid-data regimes.

As data grows abundant, the performance of joint training integrated with active object mask extraction exhibits variations across three distinct settings. In the square nut assembly scenario, joint training exhibits marginal underperformance relative to the baseline, whereas this underperformance becomes significantly more pronounced in the pick can setting with shadow. In contrast, for the square nut assembly setting with shadow, joint training with active object mask extraction consistently outperforms the baseline in data-rich regimes.

The observed performance variations across the three settings can be attributed to the interplay between policy learning and representation learning via the auxiliary task.

- **In data-rich regimes for the standard square nut assembly and pick can with shadow scenarios**, the baseline model already captures sufficient task-relevant features. Consequently, introducing the auxiliary mask extraction task induces counterproductive gradient competition during visual encoder optimization, a detrimental effect that becomes significantly pronounced in the easier pick can task.
- **Conversely, in data-rich regimes for the square nut assembly setting with shadow**, mask extraction functions as an effective structural denoiser. While shadows introduce task-irrelevant illumination variations, the mask extraction objective compels the encoder to filter out these noises and focus strictly on active objects. This contrasts sharply with pixel-level image reconstruction, which inherently amplifies visual noise by forcing the network to model extraneous shadow details to minimize reconstruction error, thereby highlighting mask extraction's utility in isolating critical task semantics.

3) Keypoint Prediction: Results of joint training with keypoint prediction under three settings are presented in Figure 16.



(a) Square (b) Square-shadow (c) PickCan-shadow

Fig. 16: Results for joint training with keypoint prediction.

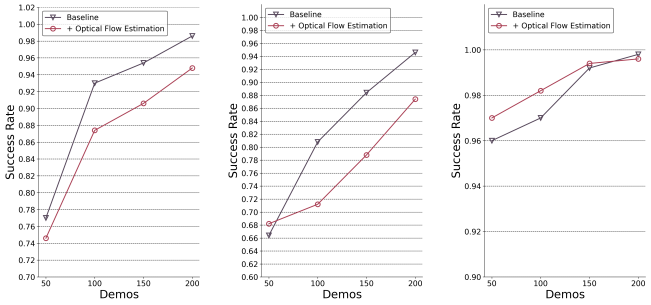
Across all three settings, the performance curves remain in the bottom-right region relative to the baseline. This indicates that joint training with keypoint prediction fails to yield sample efficiency benefits.

Figures 17a and 17b visualize a ground-truth instance and an inference example, respectively. The model generates an incorrect prediction during inference, suggesting that it fails to recognize the underlying patterns. Consequently, during evaluation, the manipulator grasps at an incorrect location, as shown in Figure 17c; this constitutes the primary failure mode. We attribute this limitation to the fact that keypoint prediction typically demands either more structurally informative inputs (e.g., point clouds [31]) or more powerful visual encoders (e.g., Vision Transformers [10]).

(a) Ground truth (b) Inference (c) Failure mode

Fig. 17: Qualitative comparison of keypoint prediction under the square nut assembly task.

4) Optical Flow Estimation: Figure 18 presents results of joint training with optical flow estimation under three settings.



(a) Square (b) Square-shadow (c) PickCan-shadow

Fig. 18: Results for joint training with optical flow estimation.

Distinct performance trends are observed across the two manipulation tasks. In the square nut assembly task (both with and without shadows), the performance curves remain in the bottom-right region relative to the baseline, indicating no gains in sample efficiency. Conversely, in the pick can task, the performance curve shifts to the top-left relative to the baseline, demonstrating clear sample efficiency benefits.

Figure 19 illustrates a comparison between a ground-truth instance and an inference example from the square nut assembly task. During inference, the model generates an incorrect estimation, suggesting a failure to capture the underlying patterns. In contrast, Figure 20 presents the same comparison for the pick can task, where the model produces an accurate estimation.

(a) Ground truth (b) Inference

Fig. 19: Qualitative comparison of optical flow estimation under the square nut assembly task.

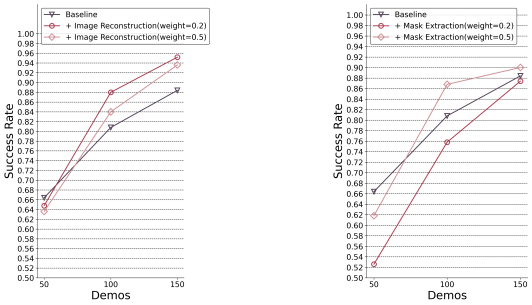
(a) Ground truth (b) Inference

Fig. 20: Qualitative comparison of optical flow estimation under the pick can task.

We attribute this discrepancy to the complexity of the manipulation tasks. Challenging tasks like square nut assembly feature highly diverse motions. While simpler tasks like pick can involve straightforward motion patterns that are inherently easier for the model to recognize.

C. Influence of Auxiliary Loss Weight

We evaluate the performance of joint training with image reconstruction and active object mask extraction under varying auxiliary loss weights within the square nut assembly task featuring shadow interference. The comparative results are reported in Figure 21.



(a) Image reconstruction (b) Active object mask extraction

Fig. 21: Comparison between different auxiliary loss weights under square nut assembly with shadow.

As illustrated, the performance of the model trained with image reconstruction degrades as the auxiliary loss weight increases. Conversely, the model utilizing active mask extraction exhibits a performance improvement. Figure 22 compares reconstructed images under varying auxiliary loss weights to the ground truth. Although a higher weight ($w = 0.5$) improves shadow restoration, reconstructing these task-irrelevant shadows hinder policy learning.

(a) Ground truth instance (b) Inference @$w = 0.2$ (c) Inference @$w = 0.5$

Fig. 22: Qualitative comparison of image reconstruction under different auxiliary loss weights.

In contrast, Figure 23 illustrates the extracted masks across different weights compared to the ground truth. A higher auxiliary loss weight effectively guides the model to prioritize task-relevant features while mitigating the impact of visual disturbances such as dynamic shadows.

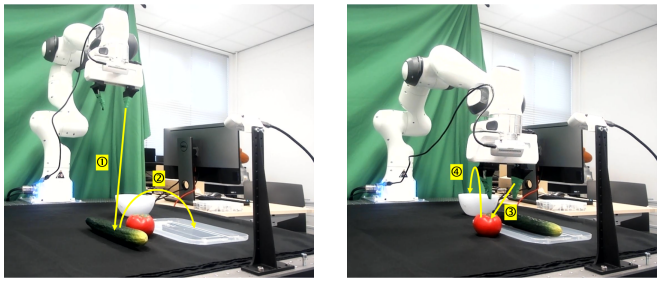
(a) Ground truth instance (b) Inference @$w = 0.2$ (c) Inference @$w = 0.5$

Fig. 23: Qualitative comparison of mask extraction under different auxiliary loss weights.

V. Real-wrold Experiment

A. Experimental Setup

As visualized in Figure 24, our real-world experiment involves a Franka arm equipped with an ARX standard gripper and two Intel RealSense D435i cameras for visual perception. One camera is eye-in-hand (wrist-mounted) to capture local details, while the other is fixed at a lateral vantage point to provide a global agent view. We designed a two-stage pick-and-place task. In the first stage, the robot grasps a cucumber and places it onto a plate. Subsequently, the second stage transfers a tomato into a bowl. The agent must autonomously evaluate whether the first stage has been successfully executed before proceeding. If the robot starts picking up the tomato before the cucumber is properly placed, the entire task will be deemed as failed.



(a) First stage: cucumber picking and placement.

(b) Second stage: tomato picking and placement.

Fig. 24: Setup for the real-world experiment.

The training demonstrations were collected via human teleoperation. In total, we gathered 30 demonstrations and evaluated both the baseline and the joint-training framework using 20 and 30 demonstrations to investigate whether a comparable success rate can be achieved under fewer demonstrations. Specifically, we only integrated image reconstruction and mask extraction into the joint-training framework in the real-world experiments, as they consistently outperformed the baseline in the mid-data regime across all three simulation tasks.

B. Results

Quantitative results for the real-world evaluation are summarized in Table V, where each success rate is an average of 20 evaluation trials. **The comparison reveals that joint training with mask extraction consistently surpasses the baseline under both 20 and 30 demonstrations.** This demonstrates that the approach achieves high task proficiency with fewer human demonstrations. **In contrast, integrating the image reconstruction auxiliary task leads to a performance drop, underperforming the baseline in both demonstration regimes.** This performance drop indicates that the pixel-level objective captures task-irrelevant noise in real-world environments. Such noise introduces distracting gradients when optimizing the visual encoder. Consequently, it hinders efficient policy learning when demonstration data is limited.

TABLE V: Success rates for the real-world experiment (+X indicates joint training with X).

Demo Quantity	20	30
Baseline	0.55	0.70
+ Image Reconstruction	0.40	0.45
+ Mask Extraction	**0.65**	**0.80**

To understand how joint training with mask extraction achieves a comparable success rate under fewer human demonstrations, we categorize and analyze observed failure modes during evaluation.

Five failure modes are observed during evaluation:

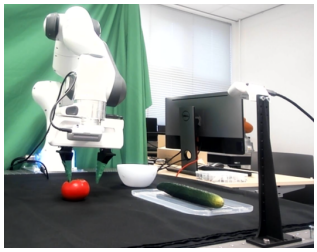
- **Gripper Stuck:** The gripper becomes physically obstructed by the tomato and cannot move further to complete the pick-up operation. (Figure 25a)
- **Execution Stalling:** The policy fails to output effective actions in certain states, causing the robot to freeze and stall the execution. (Figure 25b)
- **Off-target Execution:** The robot misses intended targets, leading to two distinct consequences. First, incorrect grasp positioning causes empty picks (Figure 25c). Second, inaccurate placement over the bowl results in dropping the tomato onto the table (Figure 25d).
- **Blind Execution:** The robot fails to complete the cucumber picking and placement. And it incorrectly transitions to the tomato-picking-and-placement stage. (Figure 25e)
- **Object Drop:** The manipulated object drops halfway. (Figure 25f)

Table VI summarizes the occurrence frequency of four failure modes for both the baseline and joint training with mask extraction. We observe two key insights:

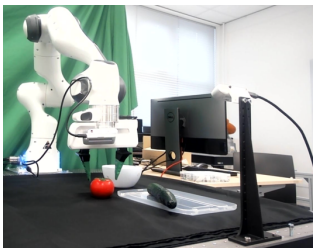
- Joint training with mask extraction achieves zero occurrences of *Off-target Execution* under both 20 and 30 demonstrations, whereas the baseline consistently suffers from this failure mode. This suggests that the introduction of active object masks provides precise spatial and geometric grounding, effectively helping the policy align with the intended target during execution.
- Joint training with mask extraction achieves zero occurrences of *Blind Execution* under 30 demonstrations, contrasting with the baseline where such failures persist. This indicates that mask extraction enhances the robot's comprehension of scene semantics, enabling it to accurately perceive whether a previous manipulation stage has succeeded before proceeding to the next.

TABLE VI: Comparison of failure mode occurrences between the baseline and joint training with mask extraction

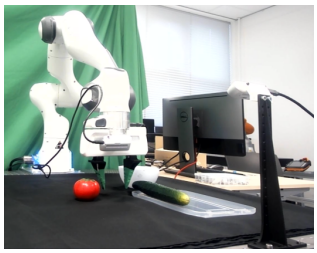
	Baseline		+ Mask Extraction	
Demo Quantity	20	30	20	30
Gripper Stuck	1	1	0	1
Execution Stalling	5	1	3	2
Off-target Execution	2	2	0	0
Blind Execution	1	2	4	0
Object Drop	0	0	0	1



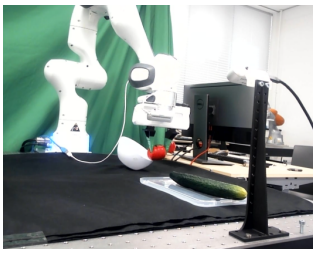
(a) Gripper Stuck



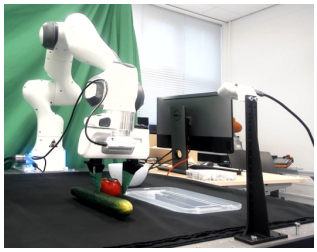
(b) Execution Stalling



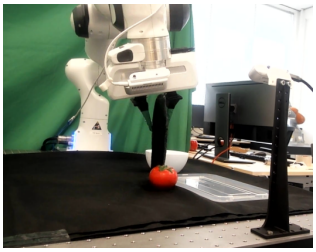
(c) Off-target Execution (incorrect grasp)



(d) Off-target Execution (inaccurate placement)



(e) Blind Execution



(f) Object Drop

Fig. 25: Prevalent failure modes encountered during evaluation.

VI. Conclusions

In this work, we comprehensively evaluated the impact of joint-training with four distinct auxiliary tasks—image reconstruction, active object mask extraction, optical flow estimation, and keypoint prediction—on robotic manipulation performance across varying data regimes and environmental complexities. Our empirical findings reveal nuanced insights into how auxiliary objectives interact with primary policy learning:

- **Image Reconstruction**: While pixel-level image reconstruction offers significant sample efficiency benefits in intermediate data regimes, its performance degrades in the presence of visual disturbances (e.g., dynamic shadows or environmental noises in real-world scenarios). This degradation stems from the objective compelling the network to focus on task-irrelevant visual details.
- **Active Object Mask Extraction**: Conversely, active object mask extraction consistently provides robust performance gains. By forcing the visual encoder to isolate task-relevant features and filter out ambient distortions, this task effectively serves as a structural denoiser that preserves semantic grounding.
- **Optical Flow Estimation**: The efficacy of optical flow estimation highly depends on the intrinsic difficulty of the manipulation task. For highly challenging tasks (e.g., Square Nut Assembly), motion patterns are difficult to capture, resulting in underperformance compared to the baseline. For simpler tasks (e.g., Pick Can), however, the straightforward motion modality allows the model to produce accurate flow estimations, thereby translating into tangible sample efficiency benefits.
- **Keypoint Prediction**: Across all evaluated settings, keypoint prediction failed to capture underlying structural patterns, yielding incorrect inference results and failing to deliver any sample efficiency benefits relative to the baseline.

In conclusion, joint training with auxiliary tasks indeed enhances sample efficiency, provided the auxiliary decoder shapes latent representations to strictly capture task-relevant information from raw observations. Specifically, this necessitates filtering out task-irrelevant environmental details while avoiding overly complex auxiliary guidance for the model to capture. These insights offer a clearer design paradigm for future auxiliary-guided joint-training or fine-tuning in robotic manipulation.

Acknowledgment

First and foremost, I would like to express my sincere appreciation to my supervisors, Dr. Cosimo Della Santina and MSc. Zhaoting Li. Throughout the course of my research, they have provided me with invaluable guidance and patient instruction. Their generous support in shaping my academic thinking and research methodology has enabled me to continuously improve and grow.

In addition, I would like to extend my sincere thanks to Zi Huang and Chenyu Zhang for their indispensable assistance during the real-world experiment.

Last but not least, I am deeply grateful to my colleagues and friends. Their help and companionship, both in research and in life, have filled this journey with warmth and strength.

References

[1] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 1–28, 2018.

[2] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[3] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 701–739, 2025.

[4] S. Nahavandi, R. Alizadehsani, D. Nahavandi, C. P. Lim, K. Kelly, and F. Bello, "Machine learning meets advanced robotic manipulation," *Information Fusion*, vol. 105, p. 102221, 2024.

[5] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[6] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg, "Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8802–8810.

[7] D. J. Foster, A. Block, and D. Misra, "Is behavior cloning all you need? understanding horizon in imitation learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 120 602–120 666, 2024.

[8] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "Data scaling laws in imitation learning for robotic manipulation," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 54 877–54 910.

[9] Y. Li and C. Zhang, "Interactive and hybrid imitation learning: Provably beating behavior cloning," *Advances in Neural Information Processing Systems*, vol. 38, pp. 47 643–47 691, 2026.

[10] M. K. Srirama, S. Dasari, S. Bahl, and A. Gupta, "Hrp: Human affordances for robotic pre-training," *arXiv preprint arXiv:2407.18911*, 2024.

[11] G. Jiang, Y. Sun, T. Huang, H. Li, Y. Liang, and H. Xu, "Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets," *arXiv preprint arXiv:2410.22325*, 2024.

[12] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*. PMLR, 2023, pp. 416–426.

[13] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 655–677, 2023.

[14] X. Li, J. Shang, S. Das, and M. Ryoo, "Does self-supervised learning really improve reinforcement learning from pixels?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 865–30 881, 2022.

[15] X. Chen, S. Toyer, C. Wild, S. Emmons, I. Fischer, K.-H. Lee, N. Alex, S. H. Wang, P. Luo, S. Russell *et al.*, "An empirical investigation of representation learning for imitation," *arXiv preprint arXiv:2205.07886*, 2022.

[16] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[17] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *Advances in neural information processing systems*, vol. 35, pp. 22 955–22 968, 2022.

[18] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on robot learning*. PMLR, 2022, pp. 158–168.

[19] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[20] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Expert intervention learning: An online framework for robot learning from explicit and implicit human feedback," *Autonomous Robots*, vol. 46, no. 1, pp. 99–113, 2022.

[21] Z. Li, R. Pérez-Dattari, R. Babuska, C. D. Santina, and J. Kober, "From action labels to sets: Rethinking action supervision for imitation learning from corrective feedback," 2026. [Online]. Available: https://arxiv.org/abs/2502.07645

[22] Z. Li, G. Chen, J. Alonso-Mora, C. Della Santina, and J. Kober, "Set-supervised diffusion policy: Learning action-chunking diffusion through corrections," in *Robotics: Science and Systems 2026*, 2026.

[23] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[24] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 995–19 012.

[25] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[26] S. Bai, W. Song, J. Chen, Y. Ji, Z. Zhong, J. Yang, H. Zhao, W. Zhou, W. Zhao, Z. Li *et al.*, "Towards a unified understanding of robot manipulation: A comprehensive survey," *arXiv preprint arXiv:2510.10903*, 2025.

[27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[28] H. Li, Z. Wang, Z.-h. Ding, S. Yang, Y. Chen, Y. Tian, X. Hu, T. Wang, D. Lin, F. Zhao *et al.*, "Robointer: A holistic intermediate representation suite towards robotic manipulation," *arXiv preprint arXiv:2602.09973*, 2026.

[29] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "Gmflow: Learning optical flow via global matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8121–8130.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[31] P. Sundaresan, H. Hu, Q. Vuong, J. Bohg, and D. Sadigh, "What's the move? hybrid imitation learning via salient points," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 51 806–51 821.

[32] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

APPENDIX

TABLE VII: Success rates for square nut assembly without shadow ($w = 0.2$ unless otherwise specified).

Demo Quantity	50	100	150	200
Baseline	0.770	0.930	0.954	0.986
+ Image Reconstruction	0.768	0.948	0.986	0.974
+ Mask Extraction	0.760	0.942	0.974	0.980
+ Optical Flow Estimation	0.746	0.874	0.906	0.948
+ Optical Flow Estimation ($w = 0.5$)	0.732	0.860	0.894	-
+ Keypoint Prediction	0.624	0.886	0.938	0.862
+ Keypoint Prediction ($w = 0.5$)	0.560	0.804	0.878	-

TABLE VIII: Success rates for square nut assembly with shadow ($w = 0.2$ unless otherwise specified).

Demo Quantity	50	100	150	200
Baseline	0.664	0.808	0.884	0.946
+ Image Reconstruction	0.648	0.880	0.952	0.942
+ Image Reconstruction ($w = 0.5$)	0.636	0.840	0.936	-
+ Mask Extraction	0.526	0.758	0.874	-
+ Mask Extraction ($w = 0.5$)	0.618	0.868	0.900	0.980
+ Optical Flow Estimation	0.682	0.712	0.788	0.874
+ Keypoint Prediction	0.438	0.610	0.830	0.854

TABLE IX: Success rates for pick can ($w = 0.2$ unless otherwise specified).

Demo Quantity	50	100	150	200
Baseline	0.960	0.970	0.992	0.998
+ Image Reconstruction	0.956	0.978	0.964	0.948
+ Mask Extraction ($w = 0.5$)	0.974	0.998	1.000	0.956
+ Optical Flow Estimation	0.970	0.982	0.994	0.996
+ Keypoint Prediction	0.870	0.932	0.974	0.976