

Exploring the Role of Foundation Models for Training Generalist Robot Learning Policies

by

Eugenia Y. Feng

S.B. Computer Science and Engineering
Massachusetts Institute of Technology, 2024

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2025

© 2025 Eugenia Y. Feng. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Eugenia Y. Feng
Department of Electrical Engineering and Computer Science
January 17, 2025

Certified by: Leslie Kaelbling
Professor, Thesis Supervisor

Certified by: Aditya Agarwal
Ph.D. Student, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Exploring the Role of Foundation Models for Training Generalist Robot Learning Policies

by

Eugenia Y. Feng

Submitted to the Department of Electrical Engineering and Computer Science
on January 17, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE

ABSTRACT

Numerous methodologies to solving goal-conditioned short-horizon tasks require hundreds of expert demonstrations, but these demonstrations are effort-intensive to collect, reducing the scalability of these approaches. Even with approaches that do work, they may have difficulty generalizing to slightly different settings. In this work, we explore two approaches to training generalist robot learning policies using large-scale foundation models.

The first approach aims to use a video foundation model to generate task-conditioned synthetic demonstrations at scale from a single expert demonstration. The objective is to leverage these synthetic demonstrations as proxy for expert demonstrations to train models that learn rewards from expert videos for solving complex visual RL problems.

The second approach seeks to improve upon the generalization ability of behavior cloning policies. Moving away from the use of videos for training, we explore using privileged representations such as keypoints or object-poses learned using open-set foundation models. By tracking pose or keypoint correspondences, the aim is to minimize the required number of demonstrations to achieve task completion and improve generalization within classes of objects.

Thesis supervisor: Leslie Kaelbling

Title: Professor

Thesis supervisor: Aditya Agarwal

Title: Ph.D. Student

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Aditya, whose guidance, encouragement, and expertise have been invaluable throughout this journey and whose mentorship has enhanced my academic and research skills. Thank you for your patience, insightful feedback, and for believing in my potential. I am also grateful for my faculty supervisor, Leslie, whose wisdom and support have been instrumental in shaping my project and inspiring my interest in robotics. Her strategic advice and thoughtful guidance have provided a strong foundation for navigating challenges and pushing the boundaries of knowledge. My thanks also goes out to the members of the Learning and Intelligent Systems group for the support and inspiration provided throughout my project. Lastly, I extend a heartfelt thanks to my friends and family for their companionship, encouragement, and support that have made this all possible.

Contents

<i>List of Figures</i>	9
1 Introduction	11
2 Initial Experiments	13
2.1 PDDL	13
2.2 AVDC	14
2.3 ACT	14
3 RL Approach	17
3.1 Motivation	17
3.2 Related Work	18
3.2.1 Video Foundation Models	18
3.2.2 Video Policy Models	19
3.2.3 Reward Models	19
3.2.4 Datasets and Benchmarks	20
3.3 Methods	21
3.3.1 Overview	21
3.3.2 Video Captioning Model	22
3.3.3 Video Prediction Model	22
3.3.4 Diffusion Reward Model	23
3.4 Results	23
3.4.1 Video Prediction Model	23
3.4.2 Diffusion Reward Model	33
3.5 Discussion and Future Work	35
4 Behavior Cloning Approach	41
4.1 Motivation	41
4.2 Related Work	42
4.2.1 Imitation Learning Policies	42
4.2.2 Keypoint Tracking	42
4.2.3 Pose Estimation and Tracking	43
4.2.4 Datasets and Benchmarks	43
4.3 Methods	44
4.3.1 Overview	44
4.3.2 Collecting Demonstrations	45

4.3.3	Extracting Keypoints and Poses	46
4.3.4	Diffusion Policy	47
4.4	Results	48
4.4.1	Collected Demonstrations	48
4.4.2	Keypoint and Pose Tracking Methods	49
4.4.3	Diffusion Policy	50
4.5	Discussion and Future Work	50
5	Conclusion	55
	<i>References</i>	57

List of Figures

2.1	A trajectory generated using PDDLStream [9] in the Kitchen Worlds [1] setting.	14
3.1	The three tasks from Adroit [19] that each video prediction model was evaluated on: (a) A manipulation task with a text prompt of "Top view of a hand opening a door in front of it"; (b) A manipulation task with a text prompt of "Top view of a hand grasping a pen"; (c) A manipulation task with a text prompt of "Top view of a hand grasping a hammer below it and using it to hammer a door to its left".	24
3.2	Results of I2VGen-XL [10] inference: (a) This is the initial scene image passed into the model with a text prompt of "A robot arm picks up the ring and places it on the peg"; (b) The right image is a frame from the output video when prompted to complete the assembly task from the Meta-World dataset [18].	25
3.3	I2VGen-XL [10] synthesized video for the Adroit door task [19].	26
3.4	I2VGen-XL [10] synthesized video for the Adroit pen task [19].	26
3.5	Seer [11] synthesized video for the Adroit door task [19].	27
3.6	Seer [11] synthesized video for the Adroit pen task [19].	27
3.7	Seer [11] synthesized video for the Adroit hammer task [19].	28
3.8	Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "four colored blocks are placed on top of a window frame, and the blue block is falling off the edge"; (b) This is a representative frame from the output video.	29
3.9	Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "a robotic arms reaches for the green cube"; (b) This is a representative frame from the output video.	30
3.10	Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "the red cube in the brown bin moves towards the teal cube"; (b) This is an early frame from the output video. (c) This frame occurs later in the video briefly, and a hallucinated block appears.	31
3.11	Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "the robotic hand is moving a single cube leftward, while the rest of the cubes remain aligned in the row"; (b) This is an early frame from the output video. (c) This is the last frame of the output video.	32

3.12	Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "A blue block slides leftward while the red block remains still"; (b-e) A sequence of the remaining frames, displaying the trajectory of the video.	32
3.13	Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "The red block is picked up"; (b-e) A sequence of the remaining frames, displaying the trajectory of the video.	33
3.14	TI2V-Zero [12] synthesized video for the Adroit door task [19].	33
3.15	TI2V-Zero [12] synthesized video for the Adroit pen task [19].	34
3.16	TI2V-Zero [12] synthesized video for the Adroit hammer task [19].	34
3.17	Dynamicrafter [13] synthesized video for the Adroit door task [19].	35
3.18	Dynamicrafter [13] synthesized video for the Adroit pen task [19].	35
3.19	Dynamicrafter [13] synthesized video for the Adroit hammer task [19].	36
3.20	AnimateDiff [14] synthesized video for the Adroit door task [19].	36
3.21	AnimateDiff [14] synthesized video for the Adroit pen task [19].	37
3.22	AnimateDiff [14] synthesized video for the Adroit hammer task [19].	37
3.23	OpenSora [15] synthesized video for the Adroit door task [19].	37
3.24	OpenSora [15] synthesized video for the Adroit pen task [19].	38
3.25	OpenSora [15] synthesized video for the Adroit hammer task [19].	38
3.26	Frames from a trajectory generated by Diffusion Reward [8] on the Meta-World dataset [18] coffee push task.	39
3.27	Two examples of scenes from different tasks from the Language-Table [20] dataset (a) The task description in this scene is "put the green star close to the green cube"; (b) The task description in this scene is "push the yellow star towards the blue moon".	39
4.1	Frames of a trajectory that successfully completes the Robomimic transport task [21].	44
4.2	The keypoints we selected for CoTracker [28] to track on the Robomimic transport task [21]: (a) Keypoints on the grippers; (b) Keypoints on the red block; (c) Keypoints on the lid handle; (d) Keypoints on the hammer handle; (e) Keypoints on the hammer head.	50
4.3	Frames of the video produced by CoTracker [28] when asked to infer the position of the selected keypoints in Fig. 4.2 on the Robomimic transport task [21].	51
4.4	Frames of execution of low-dimensional Diffusion Policy [5] on the Push-T task.	52
4.5	Frames of execution of high-dimensional Diffusion Policy [5] on the Robomimic transport task [21].	53

Chapter 1

Introduction

Humans possess the capability to observe a single demonstration of a task and subsequently execute the task themselves, even when the environment or task description varies. Replicating this generalization ability in robotics has long been a challenge. Many approaches have been proposed to address planning and execution in short-horizon tasks. These include classical task and motion planning (TAMP) [1], video policy models [2, 3], imitation learning [4], behavior cloning [5], reinforcement learning [6], and many others. However, each of these approaches has its own drawbacks that render them either difficult to scale or difficult to generalize.

Each of these existing approaches presents distinct trade-offs associated with the specific challenges they must address. Classical TAMP relies on many assumptions that are difficult to expect in a real-world setting, such as the presumption of a fully observable environment and knowledge of the state of the environment and objects in the scene at planning time. Imitation learning can be used to train models to replicate expert behavior by training on multiple expert demonstrations, but it is hard to scale, as each task requires many expert demonstrations that require substantial human effort to collect. On the other hand, reinforcement learning (RL) does not require any demonstrations, but reward design for RL agents can be tedious and requires handcrafting [7], and RL agents face challenges when

transitioning from simulation settings to real-world settings.

Video policy models [2, 3] have been used as planners that take as input an initial scene and task description, producing a video of the robot completing the task rolled out from the initial scene in the space of spatiotemporal pixels. Existing approaches plan at scale by leveraging a pre-trained foundation model as policy. Subsequently, the outputted video undergoes processing through an inverse dynamics model that infers robot actions from the pixel data to formulate the final trajectory. However, this model can be computationally expensive, and inverse dynamics does not always translate video predictions to actions well. Moreover, accurately modeling the spatiotemporal aspects of interactions remains a formidable challenge.

Imitation learning-based methods tend to overfit expert behaviors that exhibit limited diversity. To circumvent these limitations, diffusion reward models [8] learn rewards from expert videos that encourage productive exploration of expert-like behaviors. Although these methods showcase improved task diversity, they still rely on multiple expert demonstrations to model the conditional entropy of expert-like behavior, which is used as a proxy for the reward function.

Our work focuses on learning robust and generalizable policies for short-horizon tasks by exploring and evaluating each of these methods, alongside efforts to address the inherent limitations of each method. The first approach we explore is RL-based, using foundation models to generate synthetic demonstrations and task descriptions for training a reward model. The second approach is focused on behavior cloning, using an alternate representation, rather than video, in order to reduce the number of demonstrations and improve generalization.

Chapter 2

Initial Experiments

2.1 PDDL

Our preliminary investigation involved the exploration of a classical TAMP approach by setting up the Kitchen Worlds repository by the Learning and Intelligent Systems group [1]. This repository facilitates the creation of kitchen scenes in simulation and solves tasks using PDDLStream [9], a general-purpose planning framework that extends Planning Domain Definition Language (PDDL). PDDLStream supports planning algorithms with various degrees of observability. One example of a trajectory produced by PDDLStream can be seen in Fig. 2.1, where a gripper is moving a can from the counter to an open cupboard.

The PDDL domain formalization we used exhibits certain limitations stemming from its reliance on numerous assumptions, such as the presumption that the state of the environment and the objects within the scene are entirely known at the time of planning. This assumption is often impractical in real-world settings. Additionally, the translation from observation space to a PDDL specification presents a challenge, as the formulation of complete and correct scene and problem specifications can be difficult to correctly formulate. Slight modifications in these specifications can render the problem unsolvable. Consequently, we have chosen to further investigate alternative methodologies.



Figure 2.1: A trajectory generated using PDDLStream [9] in the Kitchen Worlds [1] setting.

2.2 AVDC

Next, we explored existing approaches to planning with video models, by reproducing results from the AVDC model [3], which learns through dense correspondences in actionless videos. This is a video-based robot policy created from video demonstrations without action annotations. Given an initial state in an image as well as a textual description of the goal, a diffusion model synthesizes a video of the execution. Using this generated video, the optical flow is estimated and used as dense correspondences to determine the desired transformations of the target. These transformations are then translated into robot actions.

This approach removes the need for expert demonstrations for simple tasks. However, the last step of converting from pixel to joint space is challenging as the use of an inverse dynamics model for this does not always perform well. Furthermore, there exists the issue of temporal discontinuity within the predicted video sequences.

2.3 ACT

We also explored imitation learning by evaluating the Action Chunking With Transformers (ACT) [4] model. This approach performs imitation learning using demonstrations from teleoperation. The novelty of this approach comes in the design choice to predict a sequence, or chunk, of actions instead of a single action. This is one of the leading imitation learning policies. The results of this policy are more robust and accurate for trained tasks. The drawbacks, however, are that it requires many expert demonstrations to train, and this takes

large amounts of time to collect. This policy also faces challenges with limited generalization, as it tends to overfit on expert behaviors.

Chapter 3

RL Approach

3.1 Motivation

We explored the idea of training goal-conditioned policies, where the goal is defined as an open-ended natural language command. For instance, "place the cup inside the cabinet". We aim to learn such a policy using reinforcement learning (RL) approaches. Following the initial experiments, we pursued an idea similar to AVDC, creating synthetic videos of tasks using video prediction models. In our exploration of existing RL methodologies, we identified Diffusion Reward [8] as a promising framework. Diffusion Reward is a diffusion-based reward model that assigns rewards based on the negative conditional entropy, thereby encouraging the agent to explore expert-like transitions and enhancing its capacity for generalization.

Traditional RL policies often rely on handcrafted reward functions, which can be challenging to formulate correctly and require significant effort. However, Diffusion Reward enables the training of a reward model that mitigates this issue. Furthermore, this approach obviates the need for translation from pixel space to joint space, a requirement in AVDC. However, Diffusion Reward necessitates a substantial number of expert demonstrations for training the reward model. Here, the integration of a video prediction model could prove advantageous for improving sample efficiency by generating these expert demonstrations.

Despite its potential, RL is not without its limitations, including the sim-to-real gap. As the videos utilized in our approach are generated in a simulated environment—an approximation of the real world—policies trained on these videos may fail to generalize effectively to real-world settings. This challenge is mitigated by the behavior cloning approach we introduce later in this work.

Our overall objective is to extract rewards from goal-conditioned video models as proxy for reward functions for training RL policies. To extract goal-conditioned rewards, we aim to leverage video foundations conditioned on a textual task description and an initial scene image. This approach is expected to enable the model to generalize effectively to novel objects, viewpoints, environments, and tasks, thereby advancing the state of RL methodologies.

3.2 Related Work

3.2.1 Video Foundation Models

A foundation video model refers to a large-scale pre-trained model designed to generate videos in a highly generalizable manner. In this work, we leverage an open-source text- and image-conditioned video model.

Recent advancements have led to the release of several foundation video models. One notable example is I2VGen-XL [10], which specializes in image-to-video synthesis using cascaded diffusion models. Another example is Seer [11], a video prediction model that inflates pre-trained text-to-image (T2I) stable diffusion models along the temporal axis. Similarly, TI2V-Zero [12] adapts a pre-trained text-to-video diffusion foundation model by conditioning it on an input image. Dynamicrafter [13] employs the motion prior of text-to-video diffusion models and conditions the generative process on a randomly selected video frame. AnimateDiff [14] transforms T2I models into animation generators through the use of a pre-trained motion model. Lastly, OpenSora [15] is another open-source video prediction model.

Foundation video models remain an active area of research. By integrating these models into our approach, we anticipate that our methodology will continue to benefit from advancements in this rapidly evolving field.

3.2.2 Video Policy Models

A number of existing approaches utilize foundation video models to plan in pixel space, subsequently employing an inverse dynamics model to convert these plans into joint space. Both the UniPi [2] and AVDC [3] models use text-conditioned video generation models to predict future video frames based on an initial frame and a textual description of the goal or expected robot behavior. AVDC computes the optical flow, and both models then use inverse dynamics to infer object pose and corresponding robot actions.

Video Language Planning (VLP) [16] offers another approach that leverages a video model as a dynamics model. VLP employs a tree search algorithm that uses vision-language models as policy and value functions, and text-to-video models as dynamics models. Given an initial image and a task instruction, VLP generates a video plan that includes video and language specifications for task completion.

Despite their innovation, these approaches are limited by their reliance on planning in the space of spatiotemporal pixels. Translating these plans into joint actions necessitates an inverse dynamics model, which does not always perform well. Furthermore, the physics of objects depicted in video predictions is not always accurately preserved, as modeling the temporal nature of interactions of objects remains a challenging problem. These models also tend to be computationally expensive due to the high-dimensional state space of videos, with VLP [16] requiring approximately 30 minutes to generate a single video during inference.

3.2.3 Reward Models

To mitigate the challenges associated with video policy models, alternative approaches leverage video models as proxies for RL reward functions. Two notable examples are Video Prediction

Rewards (VIPER) [17] and Diffusion Reward [8]. Both methods derive rewards from a model, either a conditional video diffusion model or an auto-regressive transformer, trained on expert videos by using the video prediction likelihoods as reward signals.

Diffusion Reward, in particular, calculates video prediction likelihood based on the negative conditional entropy, encouraging alignment with the trajectory distribution of the video model. An RL agent is subsequently trained in the joint space using this reward model. This methodology alleviates the need for handcrafted RL reward functions while simultaneously bypassing the complexities of translating between pixel and joint spaces.

3.2.4 Datasets and Benchmarks

There are four datasets explored in this work, each commonly used to benchmark robot policies. These datasets provide a standardized set of tasks along with readily available demonstrations and data.

The Meta-World [18] benchmark comprises 50 manipulation tasks with three difficulty modes. These tasks are diverse but exhibit shared structure and involve everyday objects. All tasks are performed using a simulated Sawyer arm in a table-top environment.

The Adroit [19] collection of tasks utilizes a simulated 24-degree-of-freedom (DoF) robotic hand. It consists of four task classes: object relocation, tool use, in-hand manipulation, and environmental prop manipulation.

Language-Table [20] is a dataset and benchmark containing nearly 600,000 demonstrations in both simulated and real-world settings. Each demonstration includes a video and a corresponding language description of the task.

Finally, Robomimic [21] is a framework providing datasets for eight manipulation tasks. These tasks test various capabilities, such as pick-and-place operations, multi-arm coordination, and high-precision insertion and assembly.

3.3 Methods

3.3.1 Overview

Our objective is to learn a generalizable and scalable policy for short-horizon tasks by leveraging synthetic demonstrations to train the Diffusion Reward model. The key insight driving this approach is the enhancement of the Diffusion Reward model’s scalability through using synthetic demonstrations generated using pre-trained video foundation models, rather than relying on extensive expert demonstrations. By integrating RL, video models, and reward models, this work aims to address the limitations of existing methodologies.

We tackle a real-world setting of learning to generate synthetic demonstrations from a single expert demonstration. To achieve this, we utilize a foundation video model to scale the creation of synthetic demonstrations. Similar to the Diffusion Reward model, we train a reward model based on the trajectory’s proximity to the distribution of these demonstrations, fully operating within the action space instead of the pixel space.

Given a single expert demonstration, dense video-captioning models are employed to generate a language description of the task. A fine-tuned video foundation model, conditioned on an initial image and a language-based task description, is then used to produce synthetic demonstrations. This approach addresses several key challenges: it eliminates the need for hundreds of expert demonstrations per task, obviates the requirement for an inverse dynamics model to translate pixels into joint commands, and simplifies reward design. By generating a proper distribution of synthetic demonstrations from a single expert demonstration, we enable scalable training in the action space.

This approach is composed of three main components: the video captioning model, the text-conditioned video diffusion model, and the Diffusion Reward model.

3.3.2 Video Captioning Model

Given an expert video demonstration of a task, our goal is to generate a dense language description of the task. An example of a caption for the task depicted in Fig. 2.1 might be "pick up the red cup on the table and place it inside the left cabinet."

One approach involves evaluating dense video captioning models [22] for producing task descriptions directly from videos. Alternatively, a second approach applies vision-language models (VLMs) [23] to individual frames, generating descriptions of the scene for each frame, which are then temporally combined to create a comprehensive description of the entire video. This dense language description is subsequently utilized as input for our video model in the next step.

3.3.3 Video Prediction Model

The objective is for this model to utilize the initial image from the expert demonstration video along with the dense language description of the task from the video captioning model to generate multiple videos of the robot rolling out the action. The aim is to generate a number of videos equivalent to the demonstrations typically required to train Diffusion Reward. The model is designed to generate a distribution of trajectories that effectively complete the specified task.

Beginning with a video foundation model, we condition it on the initial scene image and a text description of the task. We assessed the performance of six video prediction models in terms of video quality, temporal continuity, and grounding to the task command. Additionally, we explored fine-tuning the model on a language-conditioned robot interaction dataset, such as the BridgeData V2 dataset [24] or the Language-Table dataset [20], to determine potential performance improvements. Various fine-tuning approaches were investigated, including Low Rank Adaptation (LoRA) and layer freezing, to optimize performance.

Our observations indicated that existing foundation models, in their current state, are

not well-suited for generating task-conditioned robot-interaction videos, as they are primarily trained on large, general-purpose video corpora from the internet. Consequently, we also explored methods to adapt these video foundation models to better generate task-conditioned robot interaction trajectories.

3.3.4 Diffusion Reward Model

The synthetic demonstration videos output from the video prediction model can now be used to train any model that relies on using a collection of expert demonstrations as proxies for a reward function encouraging productive exploration of expert-like behaviors. In this work, we choose to employ Diffusion Reward to leverage its advantage of eliminating the need for explicit reward design. The implementation of Diffusion Reward is conducted largely in accordance with the methodology outlined by its authors [8].

3.4 Results

We began first with our analysis of the available open-source video prediction models as well as setting up and reproducing results of Diffusion Reward [8]. Overall, the currently available video prediction models conditioned on both image and text performed poorly in terms of grounding and consistency, as well as video quality. Due to these results, we were unable to pursue the approach further at this time. However, the approach remains promising as video prediction models continue to improve.

3.4.1 Video Prediction Model

We evaluated each model based on three criteria: in terms of temporal continuity, grounding to the task, and output quality. For each model, we conditioned it on an initial image and a textual description of the desired task. Each model was run on three separate Adroit [19] tasks, as illustrated in Fig. 3.1: (1) opening a door using the handle, (2) grasping a pen,

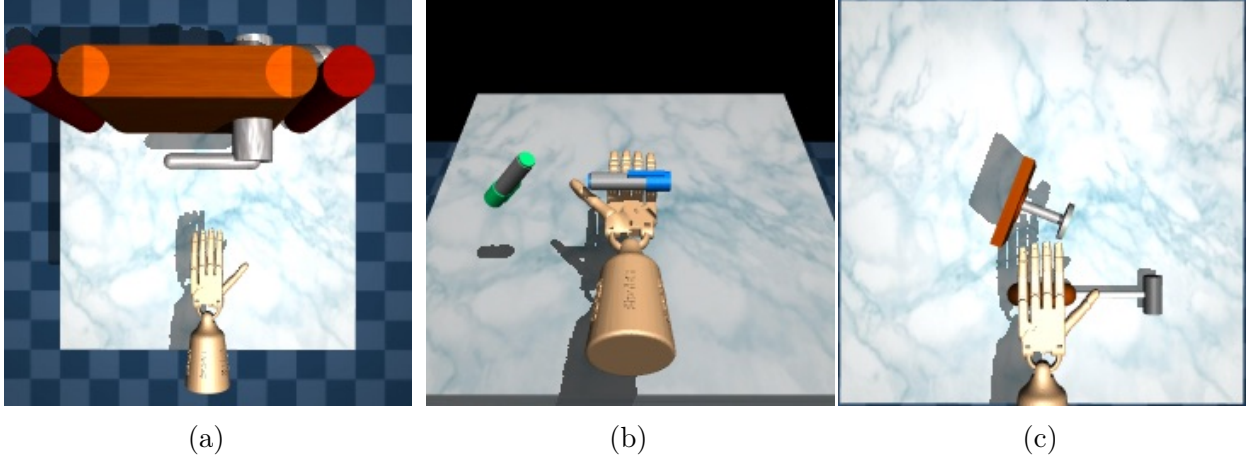


Figure 3.1: The three tasks from Adroit [19] that each video prediction model was evaluated on: (a) A manipulation task with a text prompt of "Top view of a hand opening a door in front of it"; (b) A manipulation task with a text prompt of "Top view of a hand grasping a pen"; (c) A manipulation task with a text prompt of "Top view of a hand grasping a hammer below it and using it to hammer a door to its left".

and (3) hammering a nail. Among the models tested, the Seer model demonstrated the most promising results, prompting further experimentation to validate its performance and determine whether it could support compositional generation to train the Diffusion Reward [8] model.

Additionally, we explored the application of large language models (LLMs) to enhance our task descriptions. Our findings suggest that LLMs can generate more detailed and precise descriptions, which, in turn, improved task performance in synthesized videos. In several instances, the videos generated from these refined descriptions exhibited more progress toward the task objective compared to those generated from the initial, less detailed descriptions.

I2VGen-XL

We conducted preliminary evaluations by performing inference using the I2VGen-XL video foundation model [10]. However, the model demonstrated significant limitations in temporal continuity. When prompted to generate a video in which a robot arm completes a pick-and-place task, I2VGen-XL produced a robot arm that disconnected at the joints and hallucinated the shape and action of the gripper attachment.



Figure 3.2: Results of I2VGen-XL [10] inference: (a) This is the initial scene image passed into the model with a text prompt of "A robot arm picks up the ring and places it on the peg"; (b) The right image is a frame from the output video when prompted to complete the assembly task from the Meta-World dataset [18].

An example of this inference is presented in Fig. 3.2. In this task from the Meta-World benchmark [18], the arm is required to grip the ring by the green handle and place the ring around the red block. However, in the inference results from I2VGen-XL, the gripper failed to maintain its consistency and the robot does not make progress towards the task. Due to this poor performance, we considered fine-tuning the model on a language-annotated robot interaction dataset, such as the BridgeData V2 dataset by Berkeley [24] or the Language-Table dataset [20], to potentially enhance its performance and address these issues.

When searching for a fine-tuning method for this specific model, only an archived repository [25] was available for running Low-Rank Adaptation (LoRA) [26] on the model from which I2VGen-XL is derived. LoRA is a fine-tuning method that reduces the number of trainable parameters, making it suitable for these large foundation models.

Additionally, the I2VGen-XL model is only text-conditioned and not image-conditioned, limiting its applicability for our approach. The authors of I2VGen-XL indicated that fine-tuning code would be released soon as of January 2024 [10], but no updates have been made as of February 2025.



Figure 3.3: I2VGen-XL [10] synthesized video for the Adroit door task [19].

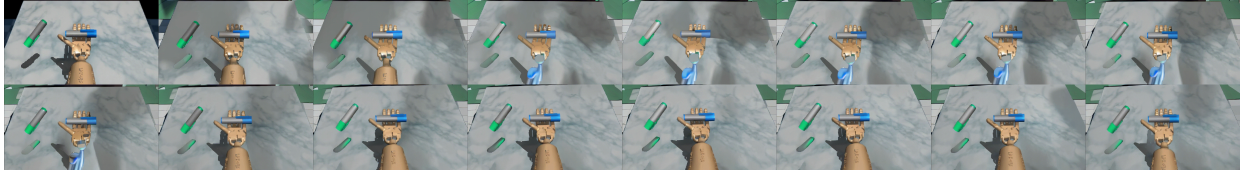


Figure 3.4: I2VGen-XL [10] synthesized video for the Adroit pen task [19].

Furthermore, when evaluating using the three Adroit tasks we tested all models on, I2VGen-XL again performed poorly in terms of temporal continuity and grounding, making no progress towards all three tasks, and creating hallucinations in irrelevant parts of the scene, as displayed in Fig. 3.3 and Fig. 3.4.

Seer

Seer [11] was among the top-performing models in our evaluations, though its overall performance was still suboptimal. While it demonstrated improved consistency in object representation and environmental settings compared to other models, it still frequently generated hallucinations. Given its relatively better performance, we proceeded to evaluate Seer’s spatial grounding capabilities on in-distribution images.

On the three Adroit benchmark tasks, the Seer model tended to hallucinate human hands and arms, rather than using the robot hand depicted in the initial conditioning image. The model also introduced variations in camera angles and did not exhibit strong temporal continuity, although it outperformed other models in maintaining consistency with the original objects present in the scene. Despite these improvements, Seer was unable to accurately depict any of the tasks being completed by the robot hand, as illustrated in Fig. 3.5, Fig. 3.6, and Fig. 3.7.

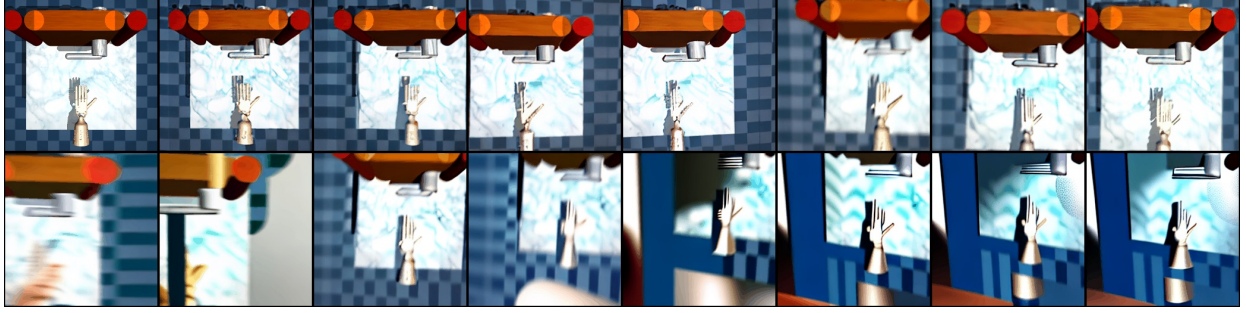


Figure 3.5: Seer [11] synthesized video for the Adroit door task [19].

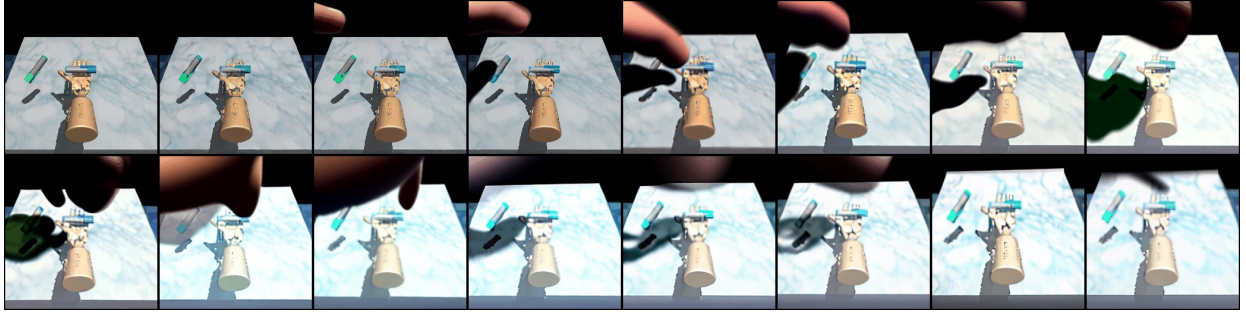


Figure 3.6: Seer [11] synthesized video for the Adroit pen task [19].

In Fig. 3.8, the initial image displays four blocks present on a window frame, and the model is prompted to display the blue block falling. However, the subsequent frames consist of images similar to Fig. 3.8b, where all four blocks remain on the window frame, with a noticeable decrease in image quality compared to the initial frame. While the temporal continuity of the objects is preserved, the task of having the blue block fall is not successfully completed.

To assess Seer’s initial performance on robot interaction scenes, we used an initial image of a robot arm above a table with blocks and a bin in Fig. 3.9. The model was prompted to generate a video of the robot arm reaching for the green cube on the table. However, in the synthesized video, a black arm-shaped object was hallucinated reaching toward the table instead of the robot arm present in the initial image. This suggests that the model fails to correctly identify the robot arm, though it still makes some progress toward completing the task.

In the example presented in Fig. 3.10, the model was prompted to generate a video of

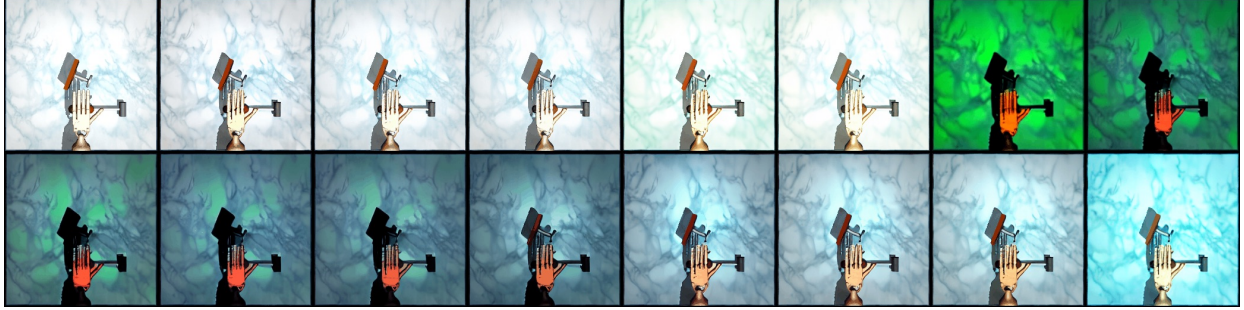


Figure 3.7: Seer [11] synthesized video for the Adroit hammer task [19].

the red block in the bin moving towards the teal cube. Given the presence of multiple teal cubes, the task could be completed successfully by moving the red block towards any of them. However, in the output video, the red block in the bin remains stationary, and a hallucinated red block briefly appears in the scene. This hallucinated block does not move toward any of the teal cubes. The model’s failure to recognize the existing red block in the bin mirrors its inability to identify the robot arm in the previous example.

In Fig. 3.11, the initial image depicts a line of cubes, with the leftmost cube already grasped by a gripper. The text prompt instructs the model to move a single cube leftward while leaving the rest of the cubes undisturbed. However, the synthesized video shows the gripper moving the cube to the right, while all but one of the cubes disappear from the line. This example demonstrates that, although the gripper maintains relatively good temporal continuity, the model fails to correctly execute the task as described in the prompt.

In the next example, we omit the gripper to further evaluate the model’s grounding. Using an initial image of two blocks, as shown in Fig. 3.12, and the text prompt instructing the model to slide the blue block to the left, we observe that the model synthesizes a hand to move a block. However, it selects the incorrect block and slides it in the wrong direction. Additionally, the blue block disappears when the red block is moved, only to reappear after the red block exits the frame. In contrast, when prompted to slide the red block to the right, the model successfully completes the task, but the two blocks briefly merge in one frame, and the blue block’s color morphs to nearly red before returning to blue.

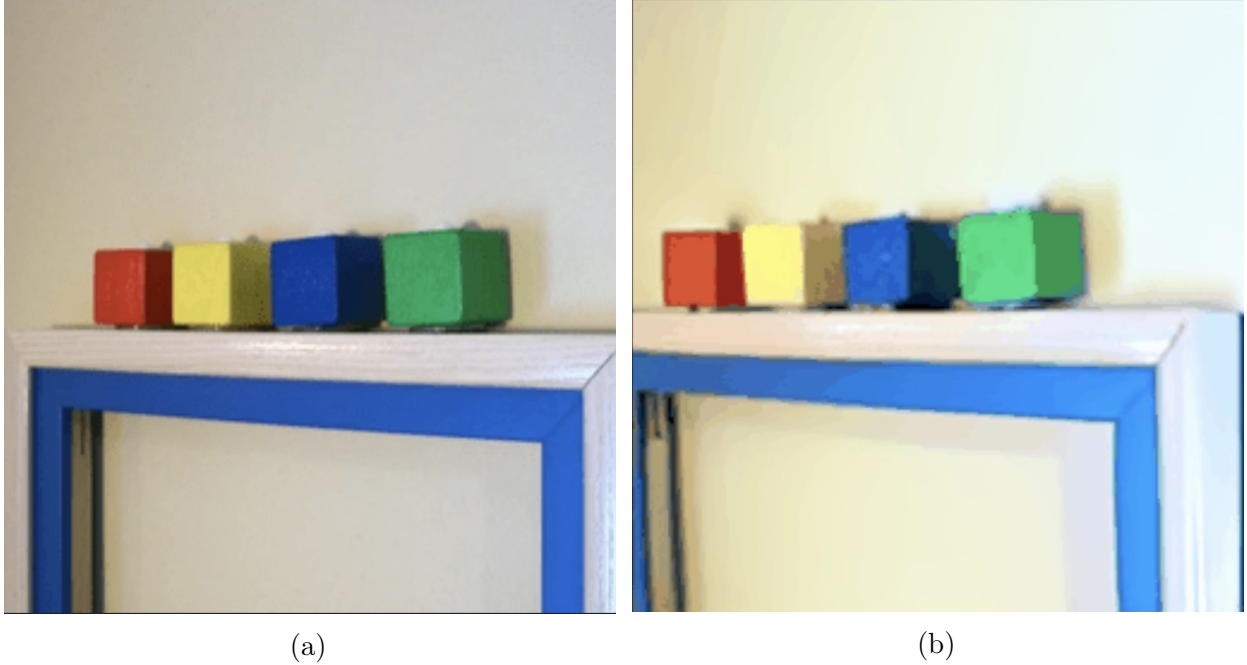


Figure 3.8: Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "four colored blocks are placed on top of a window frame, and the blue block is falling off the edge"; (b) This is a representative frame from the output video.

When the prompt is modified to remove the constraint that the unmoved block should remain stationary, the videos remain relatively consistent. The output for a prompt such as "a blue block slides leftward" closely resembles the video in Fig. 3.12, suggesting that the absence of the movement constraint does not significantly impact the model’s output.

In another test, the same initial image of a red and a blue block is used and the model is prompted to synthesize a video of the red block being picked up. The model performs relatively well, but at the end of the video, the blue block loses temporal continuity and is incorrectly transformed into a cube. When prompted to pick up the blue block, the model selects the correct object, but temporal continuity suffers, with the red block disappearing and the blue block reappearing toward the end of the video.

Next, we modify the prompt to specify that the right block should move upwards, replacing the color-based instruction with directional cues. While the videos generated from this prompt exhibit poorer temporal continuity, the model still successfully identifies the correct block to

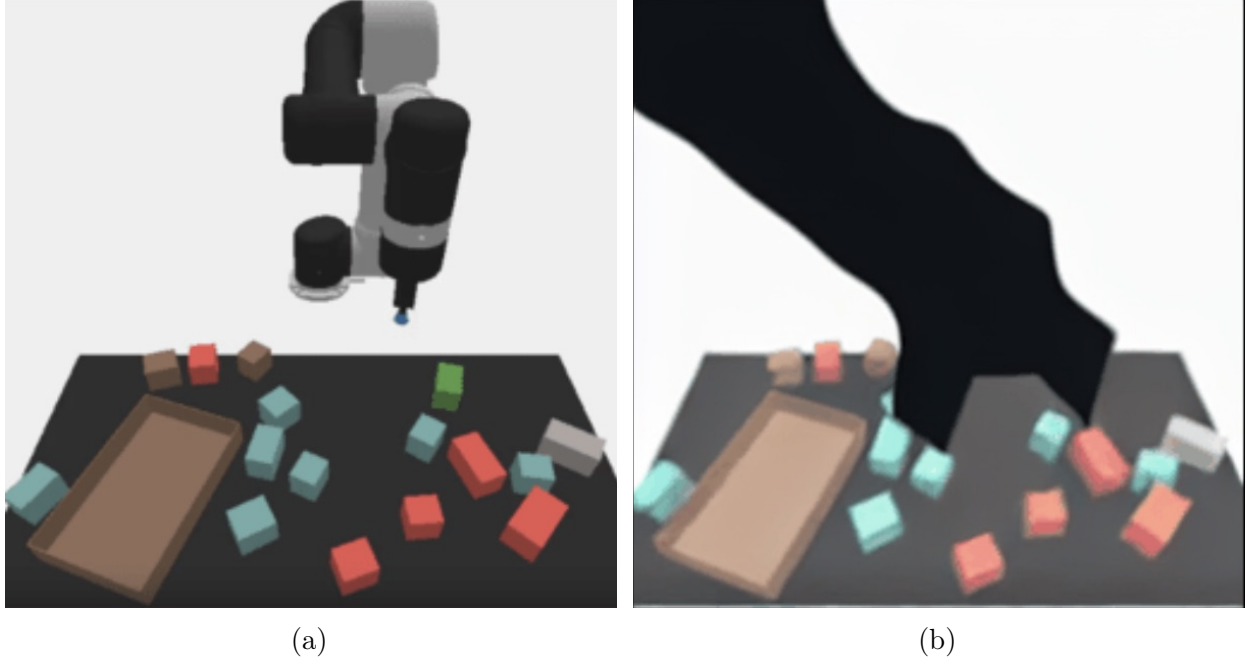


Figure 3.9: Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "a robotic arms reaches for the green cube"; (b) This is a representative frame from the output video.

move, as in the previous tests.

Overall, Seer outperforms the other video prediction models evaluated in this study, but its temporal continuity is insufficient for using the synthesized videos as reliable training data. It is possible that fine-tuning Seer on robot-interaction datasets could enhance its performance.

TI2V-Zero

In Fig. 3.14, Fig. 3.15, and Fig. 3.16, each video is conditioned on an initial frame from the Adroit dataset [19], showing a robotic hand as well as an object. In the door-opening task, the model exhibits poor temporal continuity, as the scene transitions from a door handle to a more abstract square shape. In the pen task, the video successfully synthesizes a hand grasping the pen, but the robot hand rapidly morphs into a more human-like hand. In the hammer task, the model hallucinates a human hand and generates a more conventional image of a hammer.

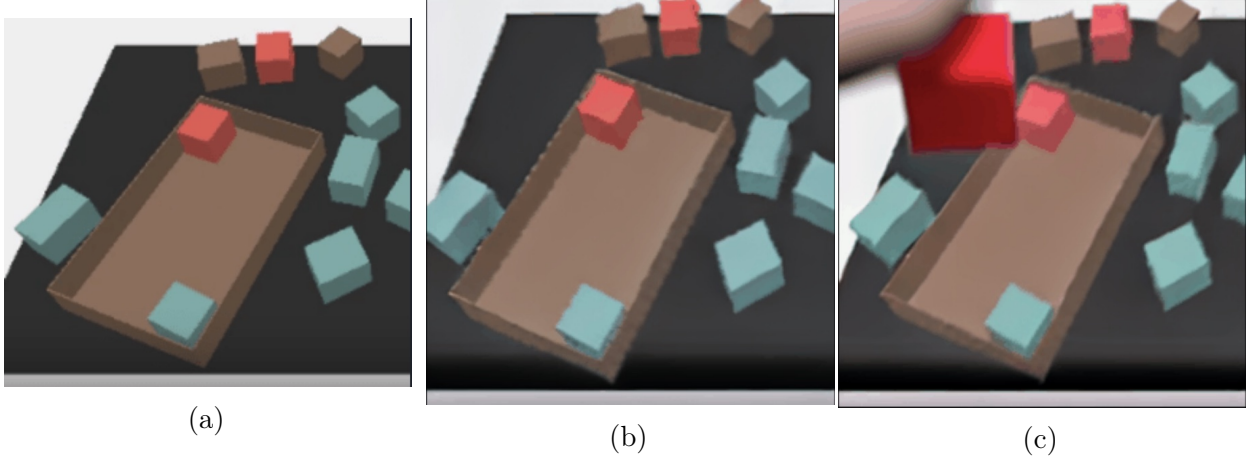


Figure 3.10: Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "the red cube in the brown bin moves towards the teal cube"; (b) This is an early frame from the output video. (c) This frame occurs later in the video briefly, and a hallucinated block appears.

Compared to Seer’s performance, TI2V-Zero [12] demonstrates weaker temporal continuity, though it appears to have relatively better grounding. Specifically, some frames remain relevant to the task, suggesting that the model maintains some level of task-relevant grounding, despite the issues with continuity.

Dynamicrafter

The model [13] performs least well on the pen task, where it hallucinates a human hand and additional objects, as shown in Fig. 3.18. In the hammer task, the hammer moves, creating a somewhat realistic video, but the robot does not make progress toward task completion, as seen in Fig. 3.19. Based on these results, Dynamicrafter demonstrates relatively better grounding than the other models, with varying degrees of temporal continuity across different tasks.

AnimateDiff

AnimateDiff [14] generates some of the most extreme hallucinations among the models evaluated, as evidenced in Fig. 3.20, Fig. 3.21, and Fig. 3.22. The model exhibits poor

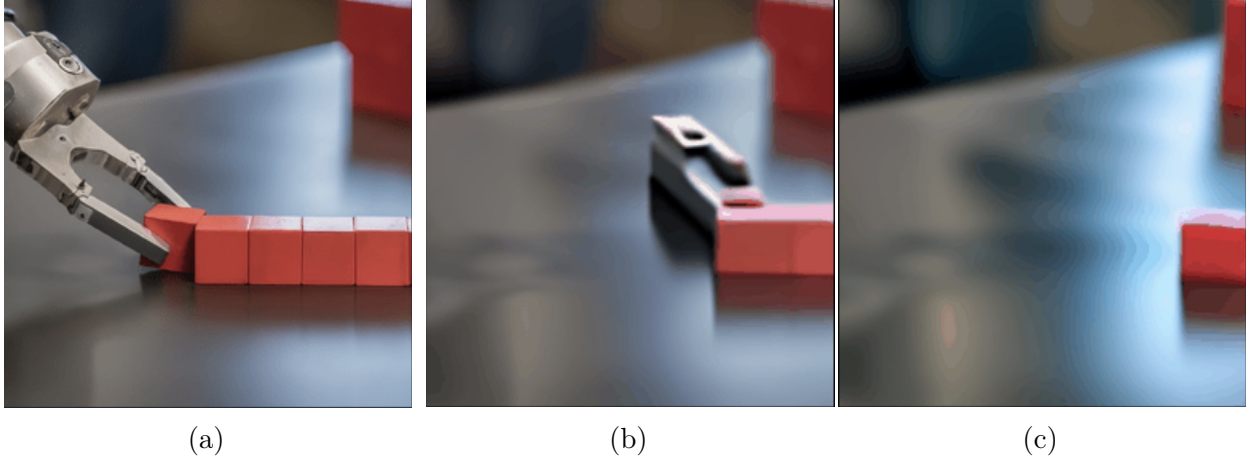


Figure 3.11: Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "the robotic hand is moving a single cube leftward, while the rest of the cubes remain aligned in the row"; (b) This is an early frame from the output video. (c) This is the last frame of the output video.

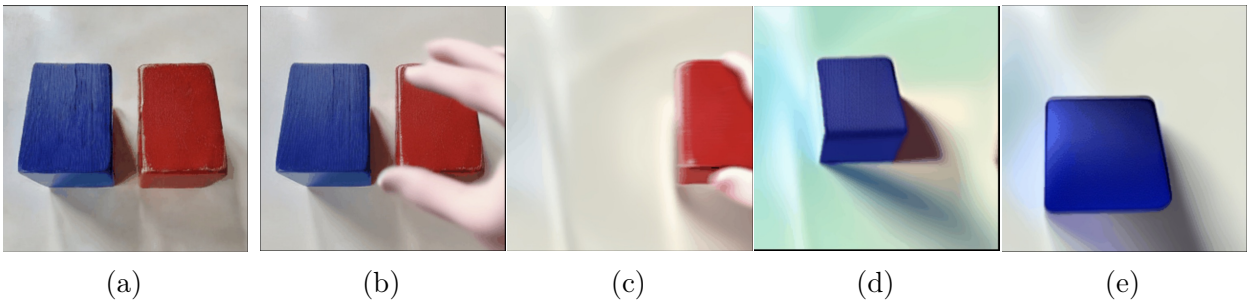


Figure 3.12: Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "A blue block slides leftward while the red block remains still"; (b-e) A sequence of the remaining frames, displaying the trajectory of the video.

grounding, as it fails to identify any of the objects pertinent to the task, and suffers from poor temporal continuity, as none of the initial objects maintain their consistency throughout the video.

OpenSora

OpenSora [15] demonstrates varying levels of success across the Adroit tasks. On the door task, the video exhibits a loss of temporal continuity, with the entire scene disappearing, as shown in Fig. 3.23. In the pen task, the video successfully displays a hand holding a pen, indicating some understanding of the task; however, it fails to ground the task within the

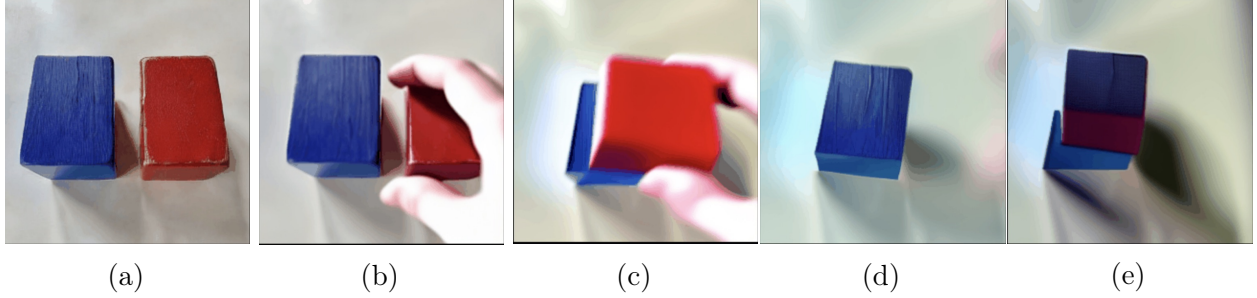


Figure 3.13: Results of Seer [11] inference: (a) This is the initial scene image passed into the model with a text prompt of "The red block is picked up"; (b-e) A sequence of the remaining frames, displaying the trajectory of the video.

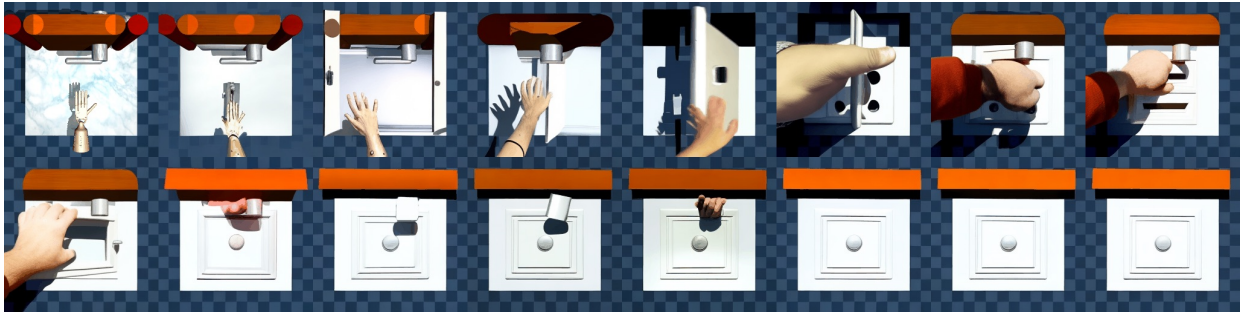


Figure 3.14: TI2V-Zero [12] synthesized video for the Adroit door task [19].

context of the initial scene, as seen in Fig. 3.24. Finally, in the hammering task, as depicted in Fig. 3.25, the model generates a video that does not advance toward task completion, predominantly producing frames that resemble the initial frame.

3.4.2 Diffusion Reward Model

We reproduced the results from the Diffusion Reward paper [8] on both the Meta-World and Adroit datasets. An example of the policy trained by Diffusion Reward is shown in Fig. 3.26 on the coffee push task.

Additionally, we sought to benchmark the performance of Diffusion Reward on the Language-Table environment [20], a suite of human-collected and annotated datasets developed by Google. Using 25 training and five test expert demonstrations across seven tasks, we observed that the video prediction quality in this setting was inferior to that in the Meta-World and Adroit tasks. Furthermore, we observed that the model struggled to generate

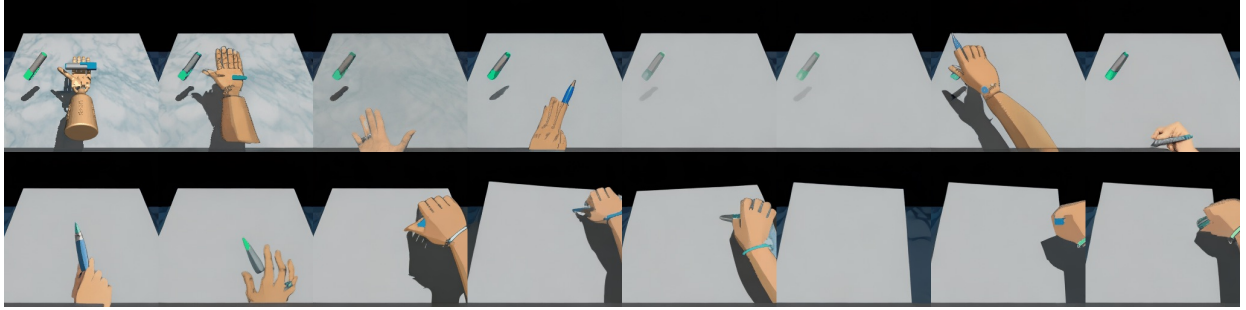


Figure 3.15: TI2V-Zero [12] synthesized video for the Adroit pen task [19].

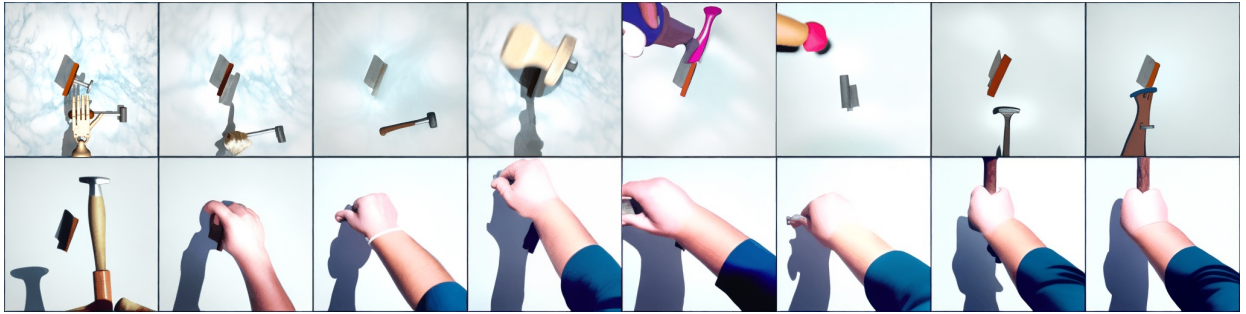


Figure 3.16: TI2V-Zero [12] synthesized video for the Adroit hammer task [19].

task-consistent predictions for tasks in scenes that closely resembled those of other tasks.

Based on these findings, we hypothesize that the environmental features in the selected Meta-World and Adroit tasks provide sufficient distinction between tasks, allowing the model to inherently capture the intended task from environmental cues. For instance, in the Meta-World coffee push task, the presence of a coffee cup and coffee machine clearly defines the task. In contrast, the Language-Table tasks often lack such distinguishing features, as many tasks may share a similar initial scene configuration. In Fig. 3.27, two different tasks from the Language-Table dataset are displayed, but their initial scene images contain all of the same objects, so it is difficult to capture the intended task description inherently from environmental cues.

Between tasks, we require the ability to differentiate either in the space of the scene or in using a language description. However, at this stage, Diffusion Reward does not incorporate a language representation space, which may limit its ability to resolve such task ambiguities.

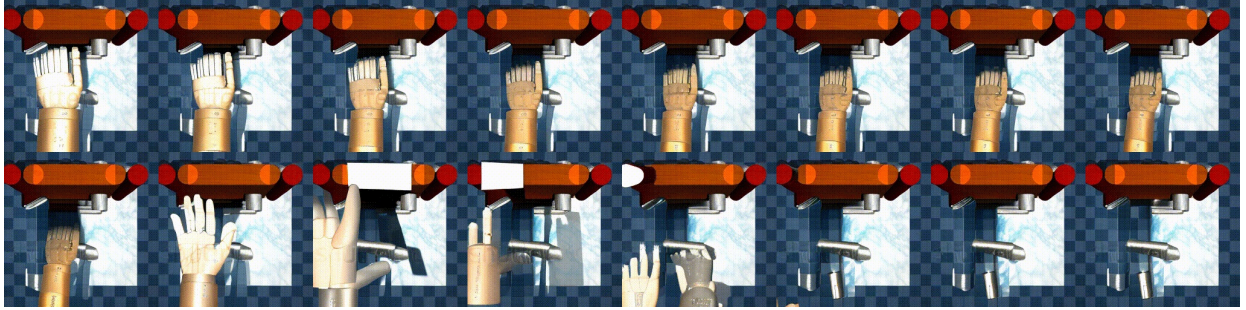


Figure 3.17: Dynamicrafter [13] synthesized video for the Adroit door task [19].

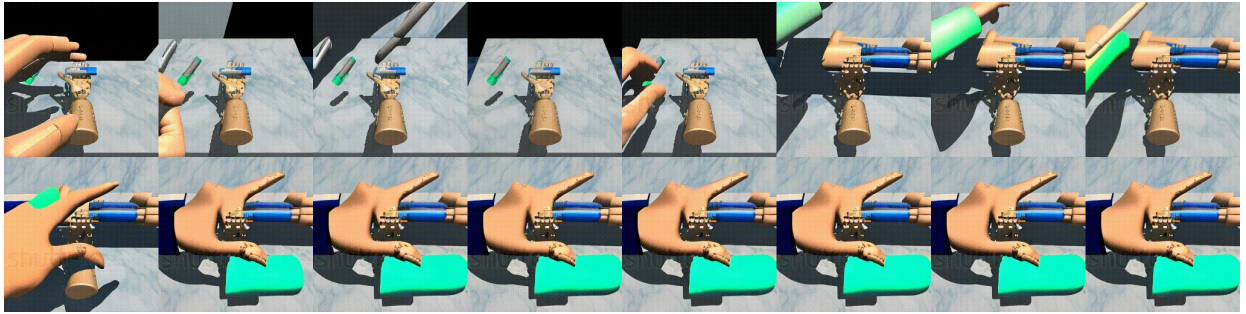


Figure 3.18: Dynamicrafter [13] synthesized video for the Adroit pen task [19].

3.5 Discussion and Future Work

Based on results from our evaluation of the six video prediction models, we concluded that the performance of current language- and image-conditioned models is not yet sufficient to make our approach robust enough. Many of these models are also designed to animate short sequences of environments, rather than a full task execution, which may account for the poor prediction performance. We will continue to monitor the progress of open-source video foundation models in future work.

In light of our problem statement and the goals of achieving a sample-efficient, generalizable approach that can learn from human demonstrations (as opposed to teleoperated demonstrations), we decided to explore alternative methods. Specifically, we pivoted towards learning from demonstrations and investigating different representations. We ultimately chose a hybrid representation, as opposed to purely visuomotor policies, with the aim of improving the generalization capabilities of the policy. This approach eliminates the need to learn a

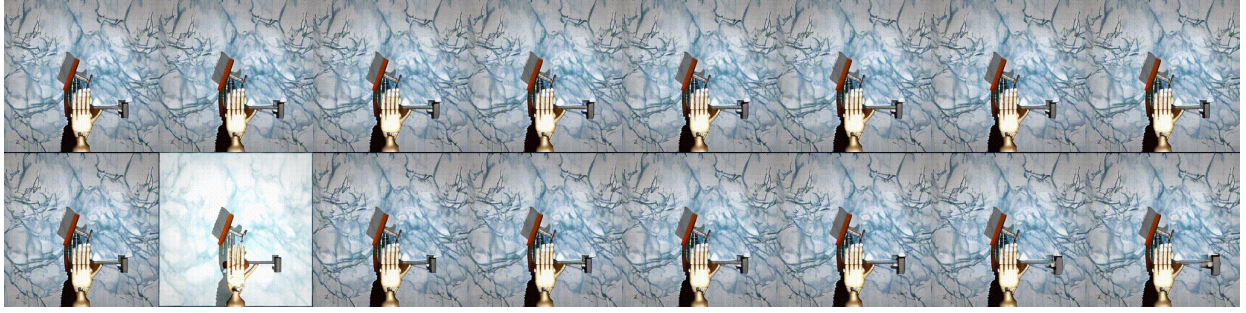


Figure 3.19: Dynamicrafter [13] synthesized video for the Adroit hammer task [19].



Figure 3.20: AnimateDiff [14] synthesized video for the Adroit door task [19].

mapping from a sequence of images to robot actions. Instead, we use the pose of objects in the image to learn the action mapping, which may be more directly learnable for the robot.

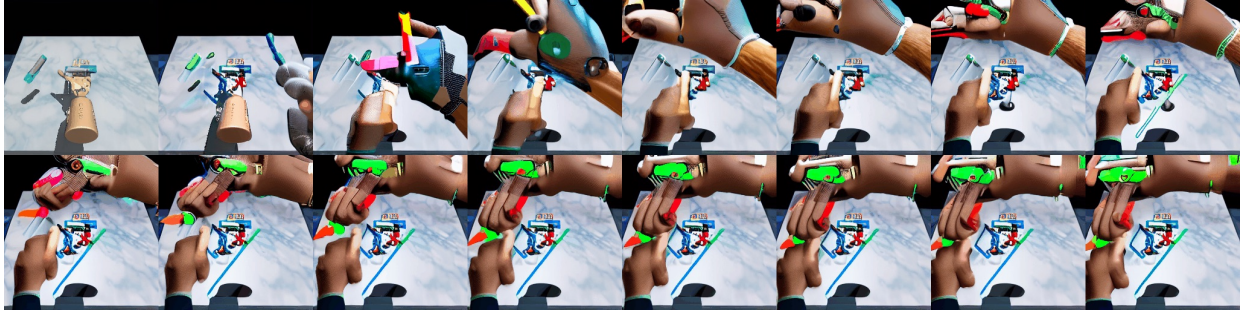


Figure 3.21: AnimateDiff [14] synthesized video for the Adroit pen task [19].



Figure 3.22: AnimateDiff [14] synthesized video for the Adroit hammer task [19].

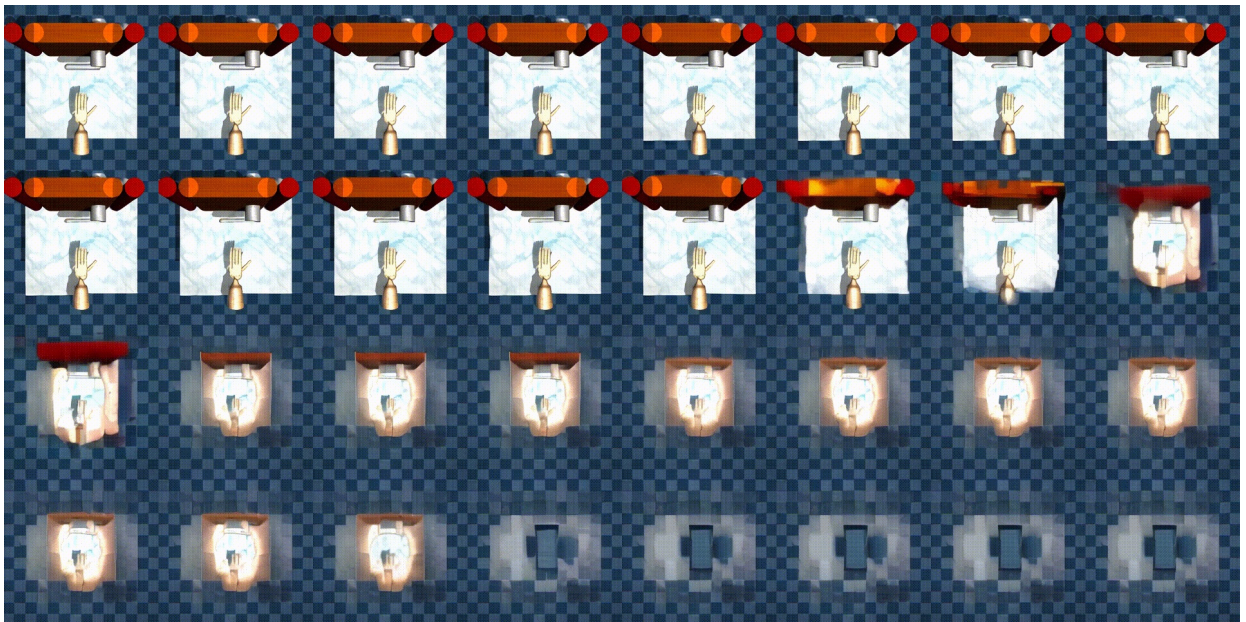


Figure 3.23: OpenSora [15] synthesized video for the Adroit door task [19].

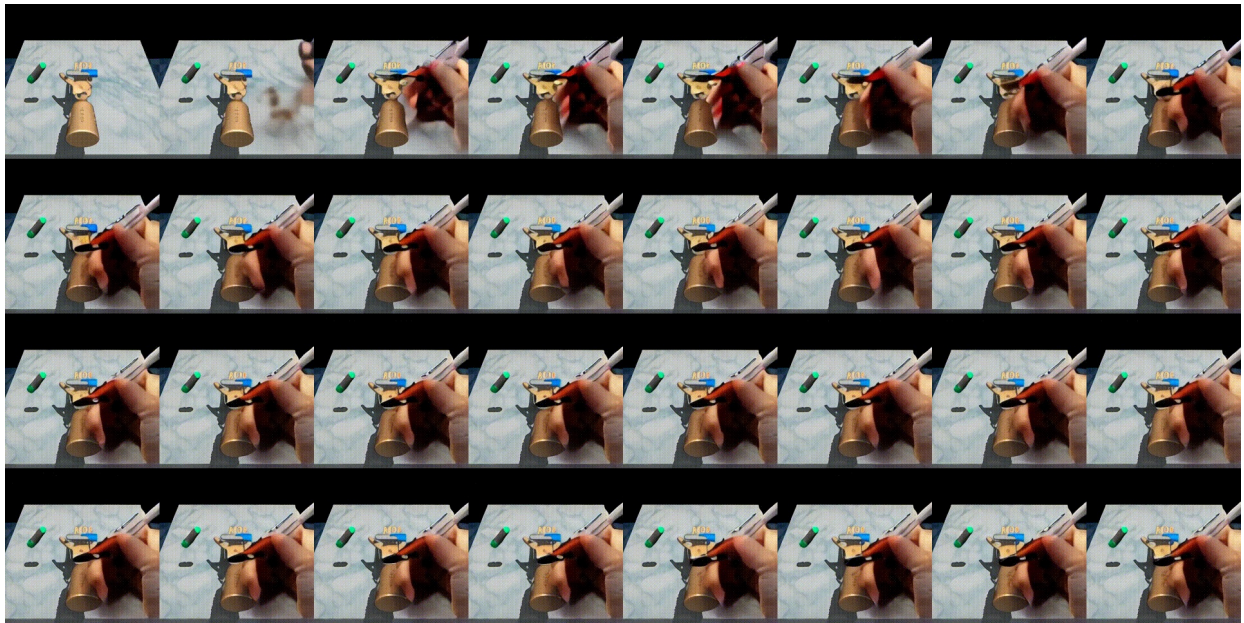


Figure 3.24: OpenSora [15] synthesized video for the Adroit pen task [19].

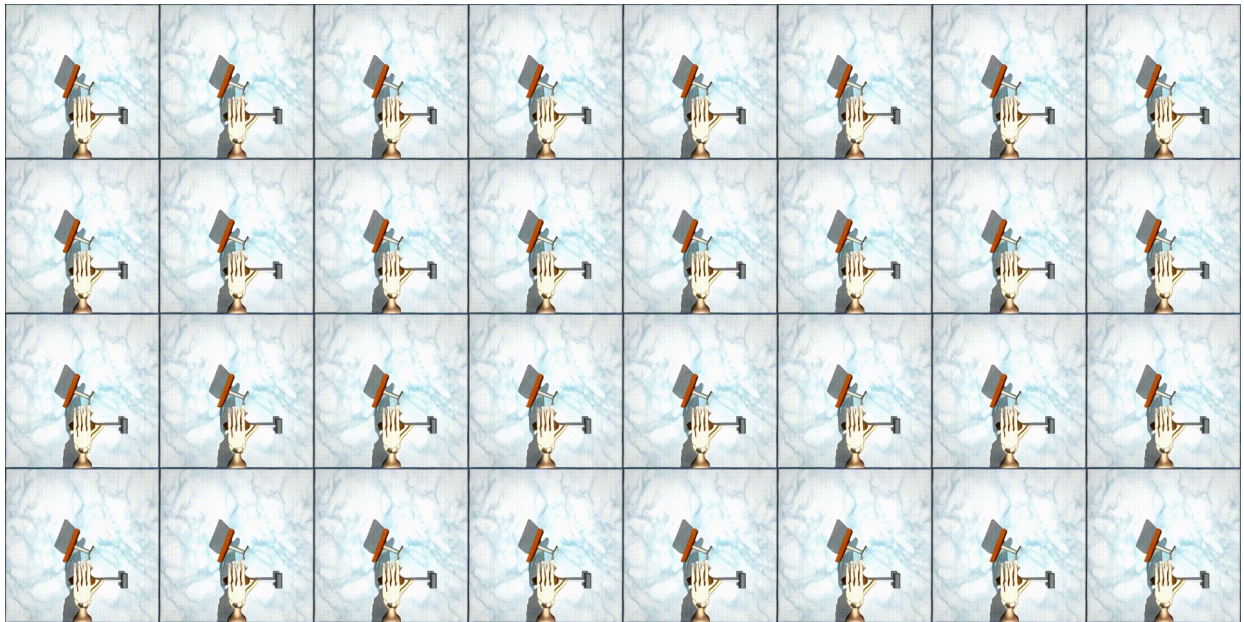


Figure 3.25: OpenSora [15] synthesized video for the Adroit hammer task [19].

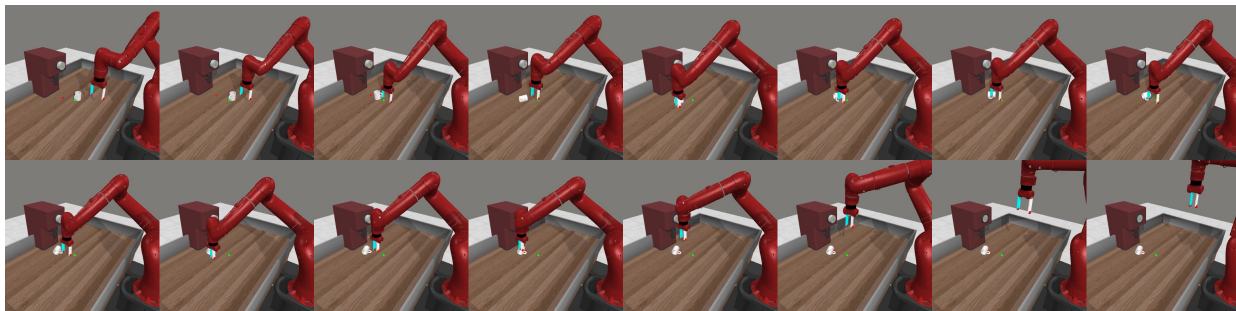


Figure 3.26: Frames from a trajectory generated by Diffusion Reward [8] on the Meta-World dataset [18] coffee push task.

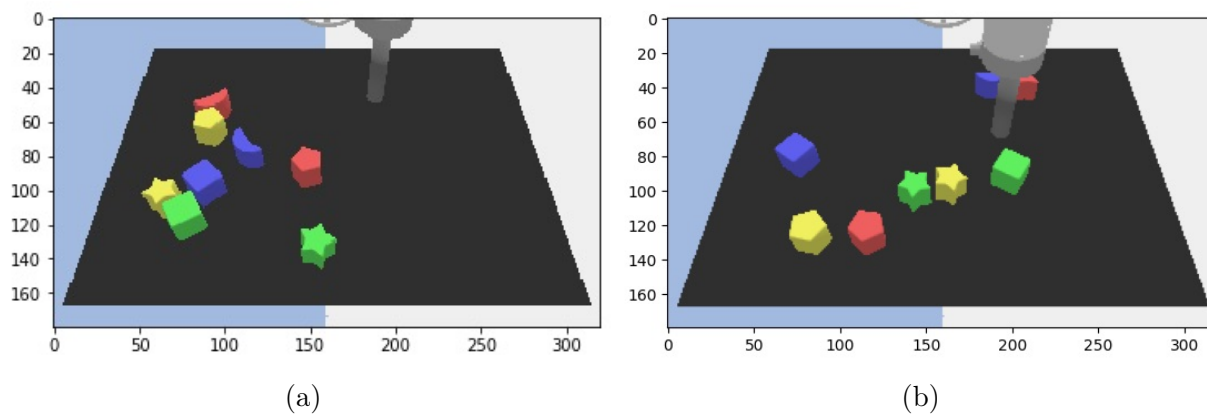


Figure 3.27: Two examples of scenes from different tasks from the Language-Table [20] dataset (a) The task description in this scene is "put the green star close to the green cube"; (b) The task description in this scene is "push the yellow star towards the blue moon".

Chapter 4

Behavior Cloning Approach

4.1 Motivation

Having pivoted to a hybrid representation using the pose of objects, we aim to apply this to behavior cloning policies to improve their generalization. With RL, the sim-to-real gap is an issue for all policies, particularly since the policy is learned in simulation. Due to this, policies trained on simulation data may not perform as expected when deployed in real-world environments.

In general, behavior cloning performs well but suffers from poor generalization and requires a large number of demonstrations. By training the policy on dense correspondences instead of images, we hypothesize that the behavior cloning model will generalize better, as there will be fewer confounding factors in the scene.

By providing the model with the change in pose of objects in the scene over time, we provide a latent space from which it can learn a policy that outputs robot actions. In contrast, with a visuomotor policy, the model would need to learn a mapping from image to latent space, followed by a mapping from latent space to robot actions. Learning the mapping from image to latent space requires many demonstrations and significant training time, as the model learns visual features this way. By eliminating this step through the use of privileged

or low-dimensional information, we expect to reduce the number of demonstrations required to train the policy.

One challenge with our approach is the requirement for action-labeled data for training. To address this, we can leverage privileged information for training, such as keypoints and poses, that can be learned from open-source foundation models [27, 28]. Additionally, to reduce the burden of collecting samples and demonstrations, we can learn from human demonstrations and videos from the internet. To improve generalization ability, we also want our policy to be embodiment-agnostic. Rather than predicting actions in the degrees of freedom of the robot, we predict the pose of the end effector and use inverse kinematics to position the rest of the robot.

4.2 Related Work

4.2.1 Imitation Learning Policies

Diffusion Policy [5] is a state-of-the-art behavior cloning policy that uses a conditional denoising diffusion process for the robot’s visuomotor policy. This approach is capable of handling multimodal action distributions and generating policies for high-dimensional action spaces. It is also robust to occlusion and perturbation and can perform well across both visual policies and state policies.

ACT [4] is another state-of-the-art imitation learning policy, as described earlier, that predicts chunks of actions. It is designed to learn policies with fewer demonstrations, making it more sample-efficient compared to traditional methods.

4.2.2 Keypoint Tracking

CoTracker [28] is a transformer-based model for keypoint tracking. It is able to track a single selected pixel or set of pixels throughout a video and is able to track these points even

through occlusions to varying degrees of success. A user is able to define a set of keypoints to be tracked throughout the entire video.

X-Pose [29] is a prompt-based keypoint detection framework. Both text and visual prompts are supported to detect keypoints. For example, if tracking a horse, the user can input an image of any horse as a prompt, which will detect and return the keypoints corresponding to horses in the input image or video. Additionally, text prompts can be used to refine these keypoints, offering further flexibility in the detection process.

ReKep [30] is an example of a work that applies keypoint tracking to solve planning. By identifying the relevant keypoints for a task, the robot is able to plan through a constraint optimization problem and manipulate soft bodies. ReKep uses DINOv2 [31] to propose keypoint candidates relevant to the task and GPT-4o [32] to generate constraints between keypoints. A constrained optimization solver then uses these constraints to return a sequence of end effector actions.

4.2.3 Pose Estimation and Tracking

FoundationPose [27] is an open-source foundation model designed for 6D pose estimation and tracking. The model supports both model-based and model-free approaches. In the model-based setting, it utilizes a CAD model of the object, while in the model-free setting, it relies on reference images. At inference time, FoundationPose can identify the 6D pose of an object and track its pose throughout a video sequence.

4.2.4 Datasets and Benchmarks

Robomimic [21] is a framework for robot learning from demonstration consisting of datasets and algorithms. The datasets consist of eight manipulation tasks testing a wide range of capabilities. These demonstrations are collected by six operators of varying proficiency. In our work, one example of a task used is called "transport," which requires multi-arm coordination. Specifically, the right arm opens a box and removes a hammer, while the left arm moves a

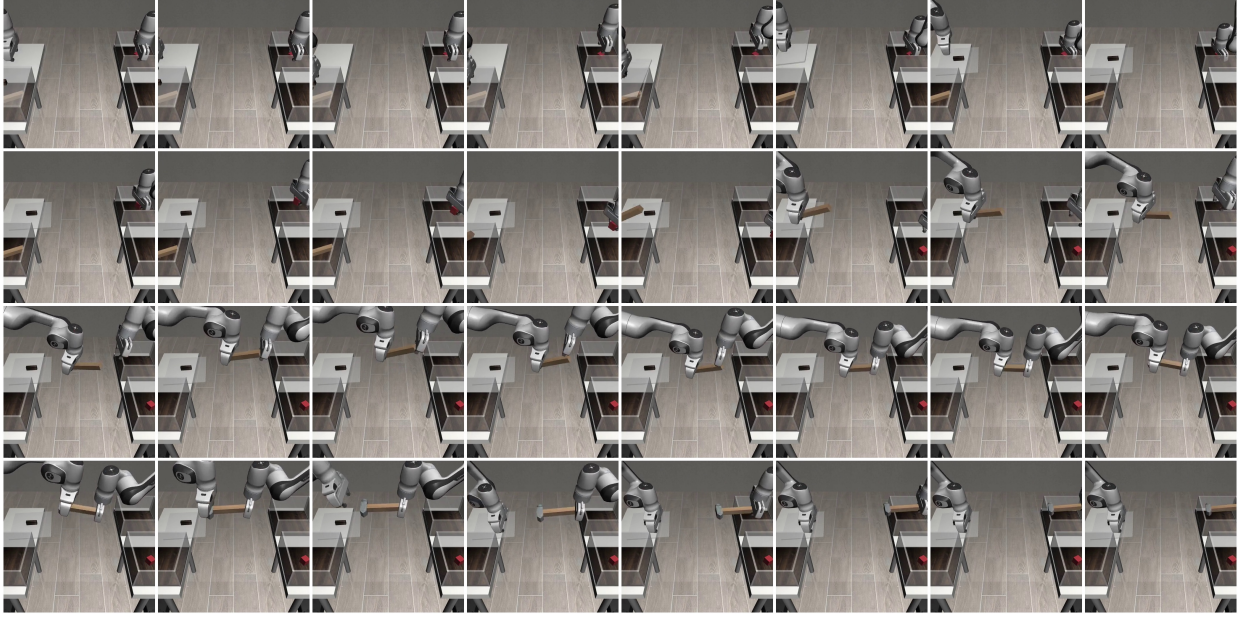


Figure 4.1: Frames of a trajectory that successfully completes the Robomimic transport task [21].

red block from one bin to another. The right arm then passes the hammer to the left arm, which places the hammer into another box. An example trajectory to complete this task is shown in Fig. 4.1.

RLBench [33] is another benchmark for robot learning, including 100 tasks that vary in difficulty. RLBench has the ability to create infinite demonstrations for each task using motion planners that take a series of waypoints specific to the task. Additionally, RLBench enables users to define custom tasks, offering a scalable approach for creating simulated demonstrations tailored to specific needs.

4.3 Methods

4.3.1 Overview

Imitation learning approaches, such as ACT [4] and Diffusion Policy [5], typically require a moderate number of expert demonstrations. With visuomotor policies, the input images are encoded into a latent space. These latent space encodings are then passed through the model

to generate a sequence of robot actions. Our approach explores using a representation of the pose of the object instead, either an estimate of the 6D pose or dense correspondences, to have the model predict robot actions. The goal is to increase sample efficiency and improve generalization by eliminating the need to learn the visual encoding.

The first approach leverages keypoints and dense correspondences. We track a number of keypoints on each object through each frame of the input video. This is likely less noisy than 6D pose estimation due to the aggregation of information from many keypoints on each object. Using this information from each expert demonstration, the model is trained on the likelihood of object trajectories within the scene.

The second approach employs pose estimation and tracking. Using FoundationPose [27], the 6D pose of objects in the scene is predicted throughout the video. This method allows for the provision of either object categories or ground truth meshes. By focusing on the pose rather than raw visual data, this approach simplifies the input representation.

We hypothesize that this approach will provide generalization within categories of objects. Unlike video-based policies, which may be influenced by unrelated visual features or interactions, this approach isolates critical spatial and structural information.

This approach can be applied to gain instance-agnostic generalization as well as allow for learning from human videos, reducing the burden of collecting teleoperated demonstrations. This will assist with sample efficiency and aid in the manipulation of articulated objects.

4.3.2 Collecting Demonstrations

To generate pose and keypoint data, we first needed to create demonstrations in a way that allowed us to extract the relevant information for each object in the scene. The first approach we explored was teleoperation, as collecting our own demonstrations in this way would allow us to access privileged information about the scene. The second approach we explored was the RLBench [33] framework, which has a method for generating simulated demonstrations.

Teleoperation

To start, we set up a kitchen scene in simulation using the PyBullet [34] simulator. The scene included a sink, a cup, and a Panda [35] arm, allowing the robot to complete the task of filling a cup with water. We initially teleoperated the robot with the keyboard, which was not very intuitive, so we transitioned to a SpaceMouse for better control. The SpaceMouse operates in six dimensions, allowing us to precisely control the position of the end-effector by moving the mouse in any direction. Next, we extracted the data of the pose of the objects in the scene, specifically the cup and the robot arm.

RLBench

We investigated the use of the RLBench framework to generate more demonstrations. RLBench enables users to define their own tasks, so we first set up their imitation learning task examples to familiarize ourselves with its capabilities, then looked into creating our own task. Following our process in the PyBullet simulator, we wanted to define a task with a goal of filling a cup with water from a sink.

4.3.3 Extracting Keypoints and Poses

Our next step was to evaluate the available open-source pose tracking and keypoint correspondences tracking methods. The most promising pose tracking method identified was FoundationPose [27], as it appears to be accurate and easy to use. For keypoint tracking, CoTracker [28] and X-Pose [29] stood out as the most applicable to our objectives.

We selected Diffusion Policy [5] as the baseline imitation learning policy to improve upon due to its adaptability. To integrate with Diffusion Policy, we saved the keypoints extracted from CoTracker and X-Pose in the same format as those in Diffusion Policy. Diffusion Policy also has a task, called Push-T, that trains on keypoints. Due to this, we can use the training data for Push-T to evaluate CoTracker and X-Pose. By passing a training video into one of

these keypoint tracking methods, we should extract keypoints that closely match those saved in the training data.

One challenge we faced was determining how to mark the initial keypoints. To be able to easily evaluate our options, we opted to mark the initial keypoints by hand. Going forward, one method to explore is tracking the keypoints of a uniform grid throughout the video and using this to identify pieces that move relative to the rest of the scene. We predict that these will be keypoints on objects relevant to the task. A second option is to use a foundation model to identify the important keypoints to track [36].

4.3.4 Diffusion Policy

We reproduced the results of Diffusion Policy, focusing on the Push-T task and the tasks from the Robomimic [21] framework. Each task was trained using two types of input representations: high-dimensional and low-dimensional. The high-dimensional input consisted of video representations of the scene, while the low-dimensional input was a vector of relevant metrics. For the Push-T task, the low-dimensional representation is a list of keypoints, while for the Robomimic tasks, the representation encompassed a variety of features, such as pose, rotation, and end effector position of the robot.

By providing keypoints or pose data, we handpick the extracted features and reduce the required space for feature extraction, minimizing the computational overhead required for feature extraction compared to training with video demonstrations. With video demonstrations, feature extraction often becomes a bottleneck, whereas handcrafted correspondences like keypoints or poses are more directly interpretable and simpler to map to robot actions.

We initially test this first on a simpler task, such as a pick-and-place task, before applying our approach to our goal of a bimanual manipulation task, namely the transport task from Robomimic. To implement this, we build upon the existing Diffusion Policy framework by defining a training method that accepts as input keypoint or pose correspondences.

Diffusion Policy’s ability to predict end-effector poses makes it inherently embodiment-

agnostic. This means the predicted pose can be adapted to different robot grippers or even human hands. Consequently, our approach has the potential to leverage videos of humans executing tasks as training data for Diffusion Policy. This increases the efficiency of demonstration collection and broadens the applicability of the model to diverse embodiments and task scenarios.

4.4 Results

We are actively refining this approach, and many of our results remain a work in progress. Our ultimate goal is to collect demonstrations for our target tasks, extract keypoints from these demonstrations, and input these keypoints into Diffusion Policy to train a generalizable policy using a minimal number of samples. Currently, each component of this pipeline is undergoing iterative improvements to enhance its effectiveness and robustness.

4.4.1 Collected Demonstrations

Teleoperation

In the kitchen scene, we experimented with teleoperation using a keyboard, mouse, and SpaceMouse. Despite the enhanced control offered by the SpaceMouse, the demonstrations created in simulation lacked smoothness and optimality in robot trajectory. Since our objective is to develop a teleoperation system that translates well to real-world settings, we decided to explore alternative methods for controlling the robotic arm to improve trajectory quality and task execution.

RLBench

In the development of RLBench [33], a novel imitation learning task was delineated with the objective of enabling the system to fill a cup from a faucet. However, the authors of RLBench caution against substituting low-dimensional task observations for image-based ones, as the

framework emphasizes the use of vision. While support for low-dimensional outputs exists, the system lacks safeguards to prevent out-of-distribution values when elements deviate from the intended area. In terms of producing a smooth trajectory that achieves the task, RLBench performs well.

4.4.2 Keypoint and Pose Tracking Methods

X-Pose

X-Pose [29] allows both text- and image-based inputs to identify objects for keypoint tracking, making it suitable for our purposes to define the relevant objects in the scene. Nevertheless, we encountered numerous setup challenges and bugs within the repository, which hindered our ability to capture a successful instance of keypoint tracking using X-Pose. Conversely, we experienced success with CoTracker, prompting us to adopt this alternative approach instead.

CoTracker

CoTracker [28] effectively tracked the selected keypoints across the video. In Fig. 4.2, the initially selected keypoints are shown on the grippers, red block, lid handle, and hammer, all objects that are required to move to complete the task. CoTracker allows selection of keypoints on multiple frames, which is helpful as it allows selection of keypoints on objects that may be occluded or out of frame during certain segments of the video. However, careful selection of keypoints is necessary to ensure good tracking performance.

Fig. 4.3 presents the results of CoTracker’s tracking. Even for keypoints selected in later frames, they are successfully tracked in previous frames, such as the hammer head that is occluded upon first entering the frame. While there are some inaccuracies in the keypoint tracking, it performs quite well overall. CoTracker is able to track occluded objects and objects that exit and reenter the frame.

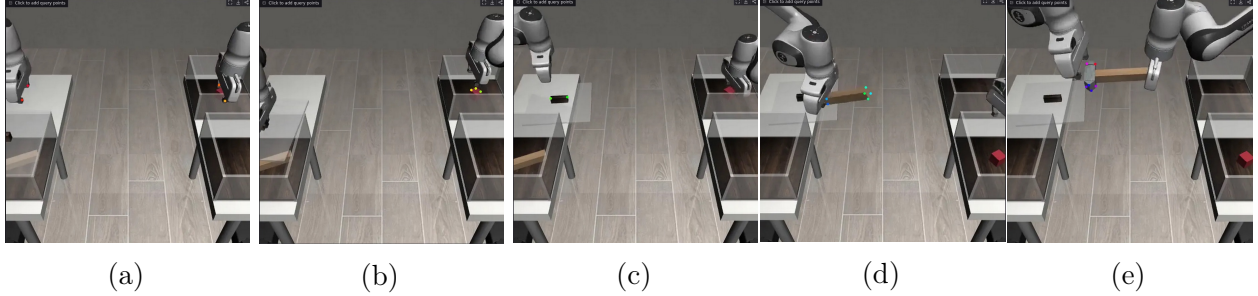


Figure 4.2: The keypoints we selected for CoTracker [28] to track on the Robomimic transport task [21]: (a) Keypoints on the grippers; (b) Keypoints on the red block; (c) Keypoints on the lid handle; (d) Keypoints on the hammer handle; (e) Keypoints on the hammer head.

4.4.3 Diffusion Policy

We successfully reproduced the results of the Diffusion Policy [5] paper. The first task for which we trained a policy was the Push-T task, using both a high- and low-dimensional representation. The high-dimensional representation consisted of videos of the task being completed, whereas the low-dimensional was a list of keypoints on the "T" shape throughout the video. The results of testing the low-dimensional policy are depicted in Fig. 4.4.

Additionally, we also successfully trained a policy for the Robomimic transport task [21] using both the high- and low-dimensional representation. In this case, the low-dimensional representation was no longer keypoints, but rather robot pose, rotation, and end effector position, the positions and quaternions of each of the objects, along with a few indicator variables. The result of testing the high-dimensional policy trained on images is displayed in Fig. 4.5. Fig. 4.1 showcases a different camera perspective, as the policy was trained on videos from multiple angles.

4.5 Discussion and Future Work

At this stage of our work, each component operates independently, and we are focused on improving each component and integrating them into a cohesive pipeline. Specifically, we are adapting the Robomimic [21] tasks supported by Diffusion Policy [5] to utilize our defined

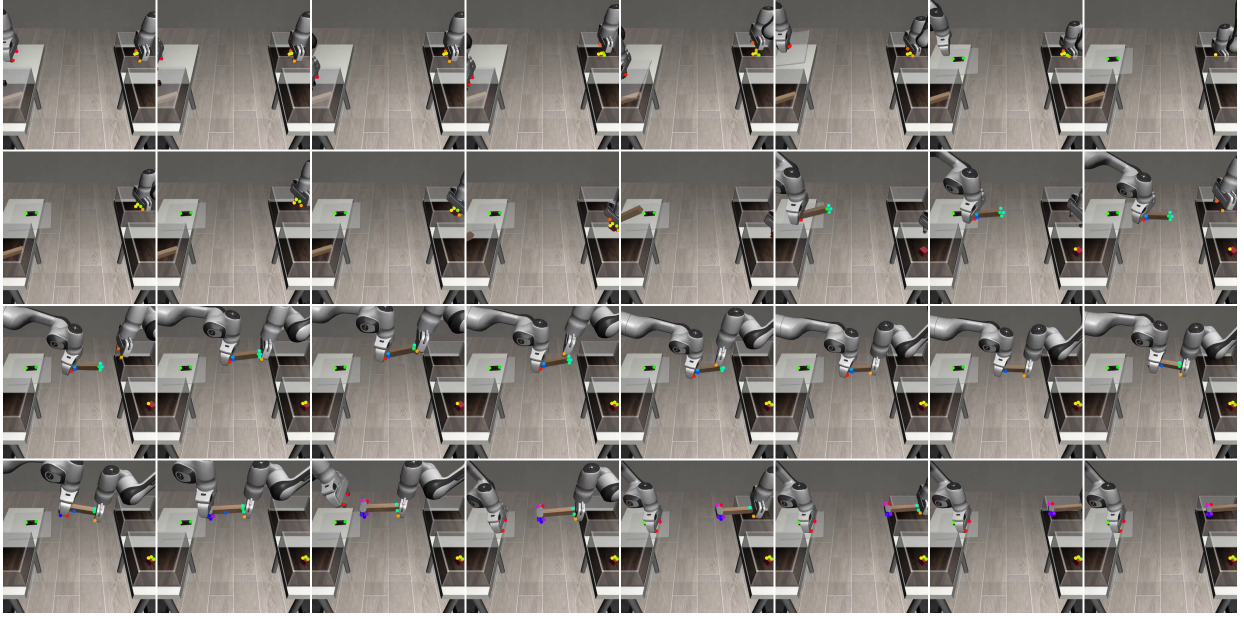


Figure 4.3: Frames of the video produced by CoTracker [28] when asked to infer the position of the selected keypoints in Fig. 4.2 on the Robomimic transport task [21].

keypoints as a low-dimensional representation encoding key task-relevant features, replacing the original representation used in the framework.

Once this adaptation is successful, we will evaluate the feasibility of using CoTracker [28] to extract the necessary keypoints from training videos. This involves mapping each training video to a corresponding list of keypoints and training the policy with this data. If the keypoints extracted from CoTracker do not suffice, we plan to pivot to using pose information as tracked by FoundationPose [27] as an alternative.

After verifying the efficacy of our representation, whether keypoint-based or pose-based, we aim to generate our own demonstrations or apply established benchmarks to evaluate the generalization ability and sample efficiency of the integrated pipeline. Additionally, we intend to incorporate human demonstrations instead of relying solely on robot interaction videos as demonstrations. With the use of keypoint or pose tracking, we can prioritize the objects relevant to the task rather than the robot itself, enabling task planning based on the trajectories of these objects.

Ultimately, our goal is to create a pipeline that integrates demonstration data, keypoint or

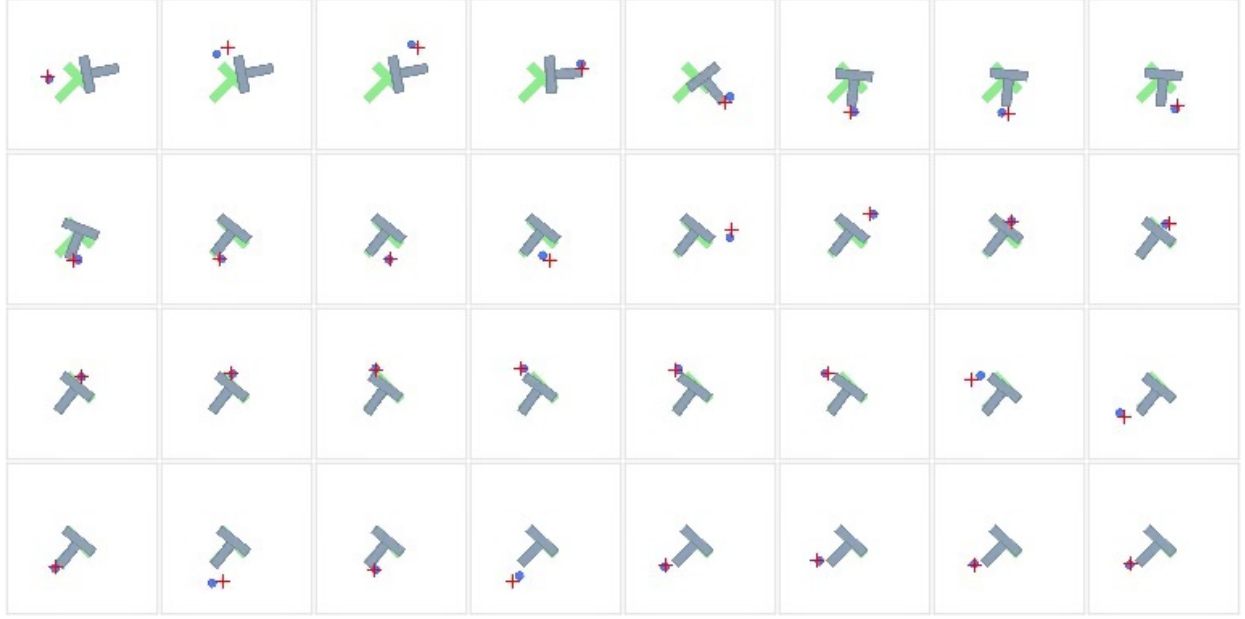


Figure 4.4: Frames of execution of low-dimensional Diffusion Policy [5] on the Push-T task.

pose tracking, and behavior cloning into a unified framework. This framework should enable the efficient generation of generalizable policies from a minimal number of demonstrations, supporting complex tasks such as bimanual manipulation and other challenging robotic tasks.

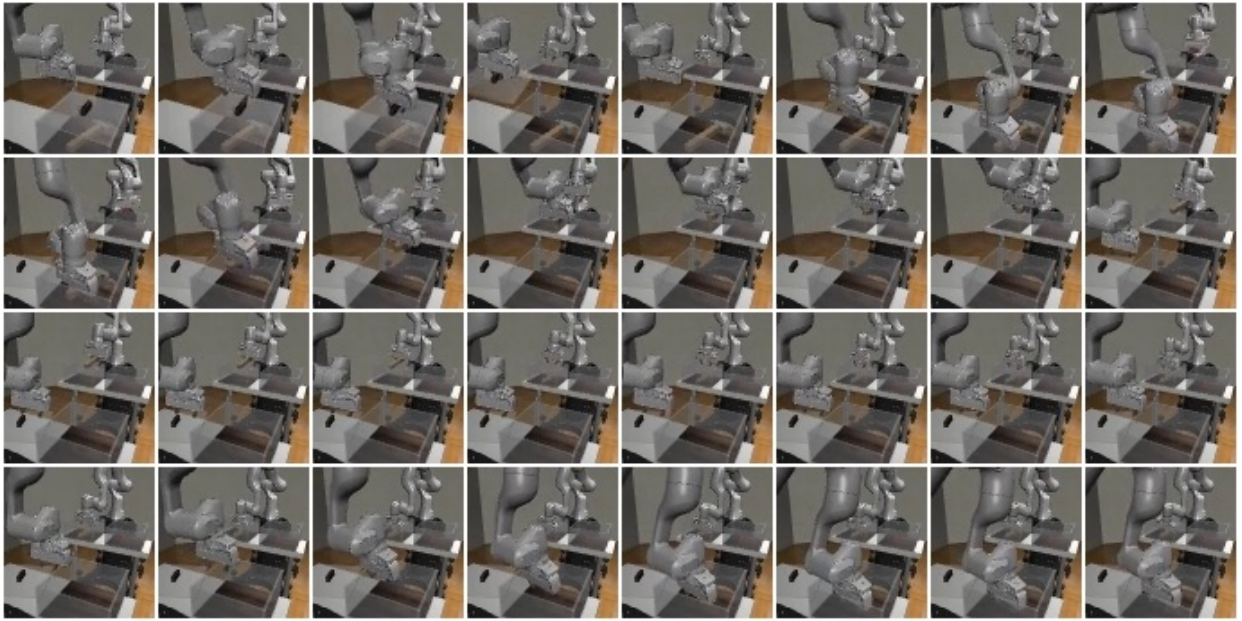


Figure 4.5: Frames of execution of high-dimensional Diffusion Policy [5] on the Robomimic transport task [21].

Chapter 5

Conclusion

In this thesis, we proposed two distinct approaches to train generalist robot policies, each designed to enhance current methodologies. The first approach utilizes RL, focused on leveraging synthetic demonstrations to train a reward model. The second approach involves behavior cloning, exploring alternate representations to improve sample efficiency and generalization.

In our first RL-based approach, we evaluated the existing open-source video foundation models, also investigating fine-tuning these models on robot-interaction datasets to improve their performance. The models overall performed poorly in terms of temporal continuity and grounding to the task, though the one that performed best in our evaluation was Seer [11].

Our objective was to use the negative likelihoods of a text-conditioned video model as a proxy for reward, rather than the likelihood of video models by having them first train on expert demonstrations. By eliminating the need for expert demonstrations, the whole training pipeline would become more sample efficient. However, the current state of open-source video foundation models did not prove sufficient to work as a proxy for reward.

In the behavior cloning approach, we investigated the use of keypoint correspondences or pose tracking as a representation, removing the need to learn the visual encoding. Pre-trained foundation models are used to identify these correspondences. We collect demonstrations, extract these keypoints and poses, and use this information to train Diffusion Policy [5],

a behavior cloning policy. This policy outputs end-effector position, allowing it to be embodiment-agnostic so that it generalizes to any robot and can learn from human demonstrations.

Future work includes revisiting the RL approach with a more temporally consistent and better grounded video foundation model. Another route is to fine-tune more of the models currently available using robot-interaction datasets and evaluate if the performance improves. For the behavior cloning approach, we are in the process of validating the impact of a keypoint or pose tracking representation on training a behavior cloning policy. Going forward, we aim to further refine these methods and explore additional avenues for advancing policy learning and addressing the limitations of existing approaches.

References

- [1] Z. Yang, C. Garrett, D. Fox, T. Lozano-Pérez, and L. P. Kaelbling. *Guiding Long-Horizon Task and Motion Planning with Vision Language Models*. 2024. arXiv: [2410.02193](https://arxiv.org/abs/2410.02193) [cs.R0]. URL: <https://arxiv.org/abs/2410.02193>.
- [2] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. “Learning Universal Policies via Text-Guided Video Generation”. In: *arXiv e-prints* (2023), arXiv-2302.
- [3] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. “Learning to Act from Actionless Videos through Dense Correspondences”. In: *The Twelfth International Conference on Learning Representations*. 2024. URL: <https://openreview.net/forum?id=Mhb5fpA1T0>.
- [4] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. *Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware*. 2023. arXiv: [2304.13705](https://arxiv.org/abs/2304.13705) [cs.R0]. URL: <https://arxiv.org/abs/2304.13705>.
- [5] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. “Diffusion Policy: Visuomotor Policy Learning via Action Diffusion”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2023.
- [6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. *Proximal Policy Optimization Algorithms*. 2017. arXiv: [1707.06347](https://arxiv.org/abs/1707.06347) [cs.LG]. URL: <https://arxiv.org/abs/1707.06347>.
- [7] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. “Eureka: Human-Level Reward Design via Coding Large Language Models”. In: *arXiv preprint arXiv: Arxiv-2310.12931* (2023).
- [8] T. Huang, G. Jiang, Y. Ze, and H. Xu. “Diffusion Reward: Learning Rewards via Conditional Video Diffusion”. In: *European Conference on Computer Vision (ECCV)* (2024).
- [9] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling. “Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning”. In: *Proceedings of the international conference on automated planning and scheduling*. Vol. 30. 2020, pp. 440–448.
- [10] S. Zhang, J. Wang, Y. Zhang, K. Zhao, H. Yuan, Z. Qing, X. Wang, D. Zhao, and J. Zhou. “I2VGen-XL: High-Quality Image-to-Video Synthesis via Cascaded Diffusion Models”. In: (2023).

- [11] X. Gu, C. Wen, W. Ye, J. Song, and Y. Gao. “Seer: Language Instructed Video Prediction with Latent Diffusion Models”. In: *arXiv preprint arXiv:2303.14897* (2023).
- [12] H. Ni, B. Egger, S. Lohit, A. Cherian, Y. Wang, T. Koike-Akino, S. X. Huang, and T. K. Marks. “TI2V-Zero: Zero-Shot Image Conditioning for Text-to-Video Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9015–9025.
- [13] J. Xing, M. Xia, Y. Zhang, H. Chen, W. Yu, H. Liu, G. Liu, X. Wang, Y. Shan, and T.-T. Wong. “Dynamicrafter: Animating open-domain images with video diffusion priors”. In: *European Conference on Computer Vision*. Springer. 2025, pp. 399–417.
- [14] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai. “AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning”. In: *arXiv preprint arXiv:2307.04725* (2023).
- [15] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. *Open-Sora: Democratizing Efficient Video Production for All*. Mar. 2024. URL: <https://github.com/hpcaitech/Open-Sora>.
- [16] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, et al. “Video Language Planning”. In: *arXiv preprint arXiv:2310.10625* (2023).
- [17] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel. “Video Prediction Models as Rewards for Reinforcement Learning”. In: *Neural Information Processing Systems* (2023). arXiv: [2305.14343](https://arxiv.org/abs/2305.14343) [cs.LG].
- [18] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. “Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning”. In: *Conference on Robot Learning (CoRL)*. 2019. arXiv: [1910.10897](https://arxiv.org/abs/1910.10897) [cs.LG]. URL: <https://arxiv.org/abs/1910.10897>.
- [19] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. “Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2018.
- [20] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. “Interactive language: Talking to robots in real time”. In: *IEEE Robotics and Automation Letters* (2023).
- [21] 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”. In: *arXiv preprint arXiv:2108.03298*. 2021.
- [22] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong. “End-to-end dense video captioning with masked transformer”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8739–8748.
- [23] Z. Liu et al. *NVILA: Efficient Frontier Visual Language Models*. 2024. arXiv: [2412.04468](https://arxiv.org/abs/2412.04468) [cs.CV]. URL: <https://arxiv.org/abs/2412.04468>.

- [24] H. Walke et al. “BridgeData V2: A Dataset for Robot Learning at Scale”. In: *Conference on Robot Learning (CoRL)*. 2023.
- [25] J. Wang, H. Yuan, D. Chen, Y. Zhang, X. Wang, and S. Zhang. “Modelscope text-to-video technical report”. In: *arXiv preprint arXiv:2308.06571* (2023).
- [26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).
- [27] B. Wen, W. Yang, J. Kautz, and S. Birchfield. “Foundationpose: Unified 6d pose estimation and tracking of novel objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 17868–17879.
- [28] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. “Cotracker: It is better to track together”. In: *European Conference on Computer Vision*. Springer. 2025, pp. 18–35.
- [29] J. Yang, A. Zeng, R. Zhang, and L. Zhang. “X-Pose: Detection Any Keypoints”. In: *ECCV24* (2023).
- [30] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. “ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation”. In: *arXiv preprint arXiv:2409.01652* (2024).
- [31] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193* (2023).
- [32] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).
- [33] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. “Rlbench: The robot learning benchmark & learning environment”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3019–3026.
- [34] E. Coumans and Y. Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. <http://pybullet.org>. 2016–2025.
- [35] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne-Schon, L. Hausperger, and S. Haddadin. “The Franka Emika Robot: A Reference Platform for Robotics Research and Education”. In: *IEEE Robotics & Automation Magazine* 29 (June 2022), pp. 2–20. DOI: [10.1109/MRA.2021.3138382](https://doi.org/10.1109/MRA.2021.3138382).
- [36] X. Fang, B.-R. Huang, J. Mao, J. Shone, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling. “Keypoint Abstraction using Large Models for Object-Relative Imitation Learning”. In: *arXiv preprint arXiv:2410.23254* (2024).