Generative Video Demonstrations for Object-Centric Robotic Manipulation Policies

by

Maggie Huili Yao

S.B. Computer Science and Engineering, Massachusetts Institute of Technology, 2025.

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

May 2026

Authored by: Maggie Huili Yao
Department of Electrical Engineering and Computer Science
May 15, 2026

Certified by: Leslie Kaelbling
Professor of Electrical Engineering and Computer Science, Thesis Supervisor

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

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

Generative Video Demonstrations for Object-Centric Robotic Manipulation Policies

by

Maggie Huili Yao

Submitted to the Department of Electrical Engineering and Computer Science
on May 15, 2026 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

Robotic manipulation policy learning has long been constrained by the need for large-scale, manually collected demonstration data, whether through robot teleoperation or human demonstration, and by the limited generalizability of learned policies across embodiments and environments. In this work, we propose a framework for learning object-centric manipulation policies entirely from synthetically generated video demonstrations. Given only fifteen images of the initial scene and a natural language task description, we leverage a video generation model to synthesize demonstrations of the desired manipulation behavior. We then employ foundation models for depth-extraction, single-image to 3D mesh reconstruction, and 6D object pose tracking to extract compact SE(3) object pose trajectories from the generated videos, which serve as the sole source of supervision for training an embodiment-agnostic diffusion policy. By operating in the space of object poses rather than raw visual observations or robot-specific action spaces, our approach is inherently robust to environmental variations and generalizes across robotic platforms without retraining. We evaluate our method on six real-world manipulation tasks across two robot embodiments, demonstrating that policies learned entirely from synthetic data can achieve competitive performance with no physical demonstrations.

Thesis supervisor: Leslie Kaelbling
Title: Professor of Electrical Engineering and Computer Science

Thesis supervisor: Aditya Agarwal
Title: Ph.D. Student

Acknowledgments

I'm immensely grateful towards my supervisors Aditya for always being willing to thoughtfully discuss my ideas and for his patient mentorship as well as Leslie for inspiring me to think beyond the problem at hand. Thank you to Sahit as well for answering so many of my questions when I inevitably broke something and the Learning and Intelligent Systems Group for all the wonderful conversations. I'm also incredibly appreciative of all my friends at MIT I've made these past few years for all the cherished laughter and memories.

And above all, thank you to my parents and grandparents. I have only made it this far because I have been able to stand on your shoulders.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robotic manipulation remains a longstanding challenge at the intersection of robotics and machine learning, requiring systems that can perceive and interact with the physical world in increasingly capable and autonomous ways. Humans can generalize a skill from a single demonstration across varying conditions and contexts, yet robotic systems have not achieved comparable capability. Developing effective manipulation policies requires both reliable task execution and generalization across diverse objects and environments without large-scale, labor-intensive data collection.

Toward addressing these challenges, this thesis proposes an object-centric robotic manipulation policy learned from generative video models, enabling efficient and autonomous policy acquisition with strong generalization. We investigate how complex manipulation tasks can be decomposed into modular stages, each addressed through either foundation models or learned object-centric representations.

1.1 Existing Methods and Limitations

While many solutions have been proposed, each comes with significant limitations. Task and motion planning (TAMP) approaches offer structured reasoning over long-horizon tasks but rely heavily on hand-crafted symbolic representations and struggle to apply to unstructured, real-world environments. Reinforcement learning can acquire complex behaviors through self-exploration, but demands extensive interaction, careful reward design, and often struggles to generalize. Classical imitation learning approaches, including diffusion-based policies, have demonstrated promising results in learning expressive action distributions from demonstrations, but remain fundamentally bottlenecked by the cost and scale of demonstration data collection, typically requiring large numbers of human teleoperated or scripted demonstrations

to achieve reliable performance.

Furthermore, most existing approaches couple perception and action tightly, learning to map directly from raw sensory inputs to robot actions. This limits cross-embodiment transfer and robustness to visual variations like lighting, background, or object appearance. These shortcomings collectively motivate a need for an approach that is both data-efficient and generalizable, capable of learning meaningful manipulation skills without large-scale, real-world data collection while remaining robust to the variability inherent in real-world deployment.

1.2 Characteristics of an Effective Policy

We first define what makes an effective task-specific policy. While not exhaustive, the following properties are broadly desirable in robotics:

1. **Data Collection Efficiency:** This characteristic emphasizes the speed and efficiency of data collection while reducing human effort. Different modalities offer varying trade-offs as hand-held grippers like UMI allow demonstrators to directly manipulate objects without operating the robot; VR headsets enable intuitive demonstrations through natural hand movements; and manual teleoperation via handling the robot with direct contact provides precise control but requires a more laborious process for the person operating the robot [1] [2]. The choice of modality directly impacts scalability as methods that minimize manual labor enable gathering larger, more diverse datasets efficiently.

2. **Sample Efficiency:** Sample efficiency is defined as how many demonstrations are needed to learn the objective. Simple tasks may require as few as 5 to 20 demonstrations, while complex tasks may exceed 100. Recent advances in policy learning architectures have improved sample efficiency: diffusion-based policies represent action distributions via denoising processes [3], while vision-language-action (VLA) models incorporate large-scale visual and linguistic pretraining [4], both enabling visuomotor skill learning from significantly fewer demonstrations than classical approaches.

   These gains in sample efficiency directly reduce human effort as fewer demonstrations mean less time spent on data collection and faster iteration when deploying robotic systems for new tasks.

3. **Meaningful Environment Representations:** During training of robotic policies, multiple forms of representations are leveraged to model the environment as closely

as possible. Traditionally, policy learning approaches have relied on raw RGB images, requiring learning an intermediate black box latent visual representation, and a mapping from this latent representation to robot actions. However, such representations are susceptible to encoding task-irrelevant information, such as background appearance or the presence of extraneous objects, which can impede generalization.

4. **Generality:** Robotic policies often aim to generalize across multiple dimensions, showcasing learning that extends beyond the conditions they were trained on. Common examples include adapting to shifts in the environment, handling different objects within the same category, or accommodating changes in object poses, appearance, camera viewpoints, etc.

1.3 Three Stage Policy

We introduce a structural decomposition based on the observation that many manipulation tasks, regardless of complexity, can be naturally described in terms of three fundamental stages.

1. **Approach and Grasp:** The robot executes a sequence of actions to move the gripper into a secure grasp configuration on the object of interest.

2. **Object-Manipulation:** During the intermediate phase, the robot has to move the grasped object from its starting position to its goal position.

3. **Retract and Reset:** At the last stage, the robot must set the object back down and return back to its neutral position, successfully completing the task.

1.4 Object-Guided Manipulation Policies via Generative Video Demonstrations

We propose that the approach-and-grasp and retract-and-reset phases can be handled without a learned policy, instead leveraging motion planners for free-space motion or foundation models for grasp proposal generation. The object-manipulation phase, in contrast, is learned by modeling object pose trajectories conditioned on a target pose. This decomposition naturally motivates an object-centric perspective: rather than reasoning over the full robot configuration space, we focus on the pose of the object itself as the primary representation, tracking how it evolves toward a desired goal state.

Such object-centricity yields policies that are more interpretable, transferable across embodiments, and robust to environmental variations. Building on this foundation, this thesis explores the use of generative video models as a scalable source of demonstration data for learning object-centric manipulation policies. Concretely, rather than collecting large-scale real-world demonstrations, we leverage a video generation model conditioned on an image of the initial scene and a natural language task description to synthesize demonstrations of the desired manipulation behavior.

From these generated videos, we estimate depth using an off-the-shelf depth estimation model and harness foundation models to reconstruct object meshes from a single image and perform zero-shot 6D object pose tracking. The resulting pose trajectory of the grasped object relative to the target is then extracted from each demonstration. These trajectories are then used to train an object-centric diffusion policy that operates entirely in the space of object poses. Our proposed system addresses the above characteristics of an effective policy as follows:

1. **Embodiment Agnostic:** As a result of operating entirely in the space of object poses rather than robot-specific action spaces, our policy is inherently embodiment-agnostic, enabling deployment across diverse robotic platforms without requiring platform-specific retraining or action-labeled demonstration data.

2. **Zero Manual Demonstration Collection:** By conditioning only on initial images and training directly on synthetically generated video, our approach eliminates the need for manual demonstration data collection in any form, whether through human demonstration, robot teleoperation, or simulation-based control. Furthermore, our framework reconstructs object meshes directly from a single image, removing the need for explicit scanning procedures required by other mesh-dependent approaches, and thereby significantly enhancing both flexibility and automation.

3. **Policy Learning Efficiency:** Our approach trains exclusively on object pose trajectories, yielding a compact representation that avoids the computational overhead of image-based methods and latent space mappings. This results in faster training convergence and efficient inference, enabling fast policy acquisition from fewer than fifteen demonstrations.

4. **Environmental Generalization:** By reasoning exclusively over object poses, our approach is inherently robust to environmental distractors such as background variation, lighting changes, and the introduction of novel objects into the scene.

However, as an overview, our system also has the following limitations:

1. **Video Generation Capabilities:** Since the system depends on video generation models, policy learning is limited by the model's ability to accurately simulate the task. Unlike teleoperation or human demonstration, video generation may not always faithfully reproduce real-world interactions, or may fail to do so entirely.

2. **Occlusion:** Since the system relies on an external camera for object pose tracking, occlusions caused by the camera angle or the robot itself can cause the policy to fail.

Chapter 2

Related Work

This chapter surveys the current landscape of robotic manipulation policy learning, beginning with standard imitation learning approaches before examining the role of foundation models, object-centric representations, and the emerging use of generative video models as a source of demonstration data.

2.1 Diffusion and ACT Based Policies

Two of the prevailing techniques for training task specific policies in recent years are Diffusion Policy and Action Chunking with Transformers (ACT) [3] [5]. The high-level design of these policies involves extracting image data from one or more cameras, along with the robot's joint states, to learn a compact latent representation. This representation is then used to predict a sequence of robot actions, with new observations continuously incorporated to update and predict subsequent action sequences.

Both approaches have demonstrated strong performance across a range of manipulation tasks, exhibiting the ability to capture complex behavior from teleoperated demonstration data while maintaining temporal consistency during execution. Their relative simplicity of training and strong generalization within a given task distribution have made them widely adopted benchmarks in the imitation learning community.

Nevertheless, in relation to the previously established characteristics of effective policies, diffusion and ACT based policies are often lacking. This is due to the fact that data must be tediously collected through teleoperation as a result of needing to annotate robot actions. More complex tasks typically require 50–100 or more demonstrations, and still often fail to generalize to variations in object position, camera angle, or appearance, necessitating further data collection.

With these policies, intermediate representations are learned from the raw data input, providing uncertainty as to which parts of the image and robot pose input was actually important to learning the policy. For example, the policy's intermediate representation may fail to capture variances in object pose and instead may find spurious correlations with appearance information such as table color or lighting conditions.

A further limitation of ACT, Diffusion Policy, and many of their variants [6][7][8] is that they adopt an end-to-end formulation in which a single policy is responsible for all three stages described in Section 1.3. This requires the policy to learn grasping from scratch despite it being a largely solved problem when addressed in isolation. Furthermore, because these methods learn to predict raw robot actions directly, they inherently lack cross-embodiment generalization, as the learned policy is tightly coupled to the specific kinematic configuration of the robot on which it was trained.

2.2 Foundation Models

The rapid advancement of robotics and machine learning has brought powerful foundation models offering pretrained, general-purpose representations capable of capturing rich information across diverse tasks and environments. This in turn reduces the need for extensive task-specific data collection and enables more robust generalization in robotic policy learning.

The applications of these foundation models range from using Large Language Models (LLMs) and Large Vision-Language Models (VLMs) to assist in generating high level execution plans to extracting high-level environment features with models like Contrastive Language-Image Pre-Training (CLIP) [9]. Foundation models now more than ever play a crucial role in augmenting policy learning by extracting comprehensive environmental information, thereby enabling improved generalization.

A recurring theme in recent robotic systems is the compositional use of foundation models, where multiple powerful models are combined in a modular pipeline to achieve strong perception and execution capabilities. Representative examples include SceneComplete and TiPToP [10][11], both of which adopt highly modular frameworks that leverage foundation models as interchangeable components, allowing individual modules to be upgraded independently and thereby improving overall system performance without additional training effort.

2.3 Object-Centric Policies

To overcome the limitations of the aforementioned approaches, a growing body of work has turned to object-centric policy learning [12]. Rather than conditioning on raw visual inputs, these methods extract task-relevant information directly from the objects themselves, such as neural implicit representations [13][14][15], 2D or 3D object flow [16][17][18][19], or object poses [20][21][22], yielding a compact and efficient representation that discards task-irrelevant visual information.

2.3.1 Point Tracking Representation

One popular avenue for object-centric policies focuses on tracking task-relevant points directly on the objects involved in the manipulation. While point tracking offers a compact and efficient representation, a key limitation is that identifying which points are task-relevant often requires manual human annotation. Even in approaches that automate keypoint selection through foundation models, such as KALM [18], the underlying demonstration data must still be collected through robot teleoperation or direct human demonstration. Furthermore, such methods depend on the ability to consistently re-localize the same keypoints across varying environments and object configurations, which can be unreliable under changes in lighting, viewpoint, or object appearance.

2.3.2 Object Pose Representation

A particularly relevant precursor to this body of work is SPOT: SE(3) Pose Trajectory Diffusion for Object-Centric Manipulation [21], which exemplifies the object-centric paradigm and serves as a foundational building block for the approach developed in this thesis. As the name implies, the driving idea behind their methodology is to have object-centric representations in which the SE(3) pose of the manipulated object is measured relative to the pose of a target object.

Through such a formulation, task constraints are implicitly encoded within the object trajectories themselves, removing the need for manually specified planning rules. This is achieved by collecting human demonstrations and extracting object pose trajectories via a pose tracking model. The resulting trajectories are then used to train a diffusion policy operating exclusively over SE(3) poses, yielding a compact and embodiment-agnostic representation.

Nevertheless, the approach carries notable limitations. First, it requires an explicit object mesh obtained through manual scanning, which introduces an additional burden prior to

deployment. Second, while human demonstration is less cumbersome than robot teleoperation, the need for manual data collection remains a practical bottleneck that limits scalability.

2.4 Generated Videos as Data in Robotic Manipulation

In the robotics community, there is growing interest in leveraging video generation models as a source of data for robot learning, spanning both planning-based and policy learning approaches. The promise of this direction lies in its potential to sidestep the costly and labor-intensive process of physical manual demonstration collection.

Recent work has demonstrated this potential; RoboDreamer successfully learned and executed robotic policies in simulation from data generated by a compositional world model [23]. With the advent of powerful video models capable of simulating real-world dynamics, there exists promise in developing policies that require no manual demonstrations whatsoever.

2.4.1 Video Models as Planners

One line of work treats generated videos as plans, using synthesized visual sequences to directly guide robot execution. RIGVID exemplifies this approach by generating a video conditioned on a language command and an initial scene image, tracking the relevant object pose, and retargeting the extracted trajectory onto the robot [24]. NovaFlow adopts a similar paradigm but tracks object flow from the generated video to derive a trajectory for robot replay [25].

However, both approaches share notable limitations. They rely on costly proprietary video generation models and treat video generation as a single-use module at inference time, producing a new video for each action rather than reusing previously generated outputs. This imposes substantial computational overhead, rendering real-world deployment impractical. NovaFlow, for instance, requires 8 H100 GPUs to generate eight candidate videos in parallel, resulting in inference latency on the order of minutes per rollout.

2.4.2 Inverse Dynamics Models

A related approach couples generated videos with inverse dynamics models, which infer the robot actions required to transition between consecutive frames [26]. Ko et al., for instance, synthesize videos of a robot executing a task and recover actions without any action annotations by leveraging dense inter-frame correspondences [27]. While this reduces the burden of action-labeled data collection, a fundamental limitation is that inverse dynamics

models are inherently robot-specific as they must be specifically trained to predict actions in a particular robot's action space, precluding cross-embodiment generalization.

Chapter 3

Methodology

This thesis describes a novel method for training object-centric robotic manipulation policies trained exclusively from synthetically generated demonstrations, leveraging video generation models and complementary foundation models in lieu of any real-world data collection.

The system's input–output mapping for policy training is as follows: the inputs comprise fifteen RGBD images of initial scene configurations, one of which is a stereo image pair, and a natural language instruction specifying the target task, and the output is a learned policy as demonstrated in Figure 3.1. This chapter details the underlying architecture and the technical implementation of the proposed system.

3.1 Problem Statement

3.1.1 Target Tasks

While simple pick-and-place tasks can often be addressed through classical planners, this thesis targets a more challenging class of manipulation tasks where intermediate motion constraints make planning-based approaches insufficient. An example is the pouring task, in which a pitcher must move and tilt forward to pour liquid into a cup while maintaining a specific orientation throughout the trajectory, a constraint that can be captured by tracking the changing SE(3) pose of the pitcher relative to the stationary target object. A further example is placing a mug under a coffee machine, where the desired target pose of the mug relative to the machine is learned implicitly from demonstration data, but would require explicit specification in a planning-based approach. Though the framework is broadly generalizable, this work focuses on single-arm, single-object manipulation, with dual-arm manipulation left for future work.

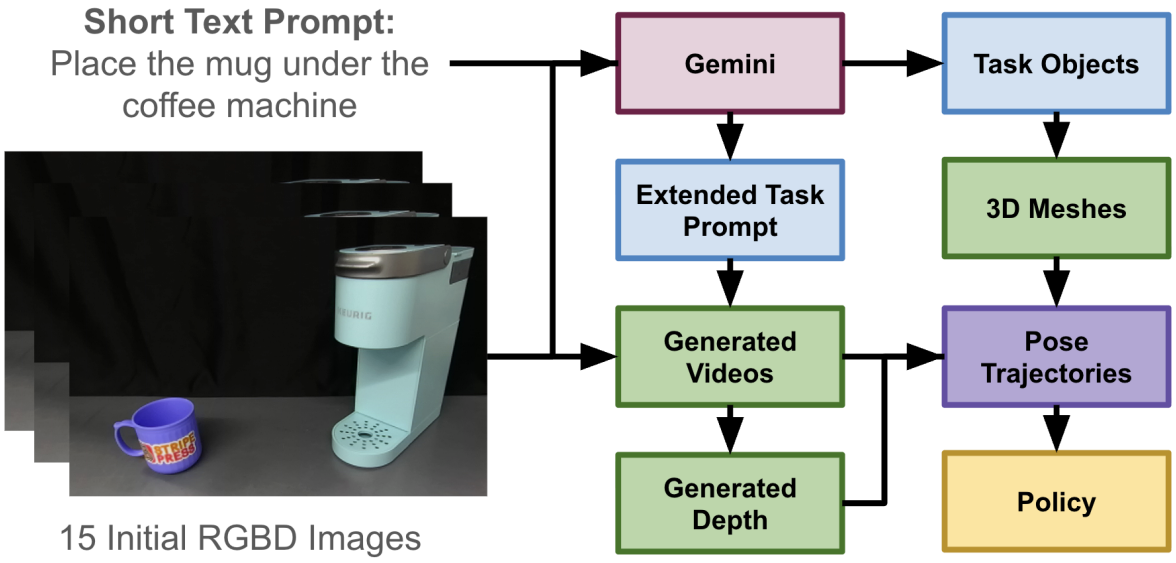

Figure 3.1: High-level architecture diagram showing fifteen RGBD images (including a stereo pair) and a text prompt as inputs, producing a trained policy as output.

3.1.2 Problem Objective

All tasks are formulated following the object-centric framework introduced in SPOT [21], consisting of a manipulated grasped object and a target object, collectively denoted as $\mathcal{V} = \{v_g, v_t\}$. Given a video demonstration of length l, the objective is to extract the full SE(3) pose trajectory of the grasped object $\{T_g^i\}_{i=1}^l$ and the initial pose of the target object T_t^1, which remains stationary throughout the task. The grasped object's pose trajectory is then expressed relative to the target object's initial pose. The resulting relative trajectory $\{\hat{T}^i\}_{i=1}^l$ serves as the training signal for the diffusion policy expressed in more detail in Section 3.4, encoding task constraints independently of the scene configuration.

3.2 Demonstrations from Video Generation Models

By conditioning a video generation model on an initial scene image, we leverage it as a proxy world model to synthesize demonstrations of the target task, thereby bypassing the need for either human teleoperation or simulation-based data collection.

3.2.1 Prompt Generation

Since video generation models require detailed prompts to minimize hallucinations, we use Gemini 2.5 Flash to automatically elaborate a short action description and single scene image into a detailed prompt, using the instructions shown in Figure 3.2. The same extended prompt is then used for all demonstrations to maintain consistent replications of the action.

A key distinction between our video generation methodology and prior works that employ generative video models for robotic manipulation is our choice of the means of manipulating the object in the generated demonstrations. Rather than depicting objects being manipulated by human hands as existing approaches do, we specify in our prompt generation that objects are acted upon by thin, semi-transparent strings descending from above, in the manner of marionette puppetry as can be seen in Figure 3.4. This design choice is motivated by the interference that visible hand geometry introduces during depth extraction that leads to inaccurate tracking.

A further advantage of using transparent strings to guide the manipulated object is that it eliminates hand-induced occlusion of the object during the manipulation phase. Moreover, it highlights a broader capability of video generation models: the ability to synthesize physically plausible motions that would be difficult or unnatural for a human demonstrator to perform. In this instance, the generated demonstrations naturally avoid occlusion of the grasped object, producing constraint-satisfying trajectories that are advantageous for policy learning.

A similar extraction logic is applied to define the two primary objects within the scene. As illustrated in Figure 3.3, Gemini 2.5 Flash identifies the grasped and target objects through relying on the extended prompt alone, outputting them in a `<basic_color> <object_type>` format. This specific nomenclature is required to disambiguate objects of the same category which is particularly important for tasks such as nested bowl stacking. The extracted object labels are later utilized to generate masks for the relevant entities, thereby facilitating accurate pose estimation within the system pipeline.

3.2.2 Video Generation with Wan2.2

We leverage the capabilities of Wan2.2, a model developed by Alibaba that is the most powerful open-source video generation model currently available [28]. Released in 2025, it represents the state of the art model relying on a Mixture-of-Experts (MoE) architecture in which a high-noise expert is responsible for establishing the global structure and spatial layout while the second low-noise expert specializes in the refinement of fine-grained visual attributes such as texture and lighting.

Prompt Generation Instructions

You are an expert at creating detailed prompts for image-to-video (I2V) generation models. Given this base prompt and reference image, generate a highly detailed extended prompt.

BASE PROMPT: `{base_prompt}`

CONSTRAINTS:

- NO hands, human figures, or body parts at any time
- Manipulated object retains identical shape, color, size, and material from first frame to last (no morphing, color shifts, or texture changes)
- All non-manipulated objects and the background remain perfectly static (no sliding, shifting, wobbling, warping)
- Static camera; smooth physically plausible motion; minimal single-axis rotations only
- Strings remain **CONTINUOUSLY VISIBLE** and taut throughout — they do **NOT** disappear during motion
- Video **ENDS** immediately upon task completion (no returning, resetting, or lowering back)

Output EXACTLY 2 SHORT PARAGRAPHS, 150–190 words total.

Paragraph 1 — String manipulation and motion (∼130–160 words):

- Begin with ONE short sentence ($\leq$25 words) that names the manipulated object (basic color + object type), the target object (basic color + object type), and the scene in a single brief phrase (e.g., "A blue ladle and a white bowl sit on a wooden tabletop."). No materials, patterns, or extra modifiers.
- Then write: "No hands, no human figures, no body parts appear at any time."
- The video opens with the scene perfectly static. Thin semi-transparent strings first descend visibly from the top of the frame and make contact with the manipulated object at specific points (state number and locations). No strings touch any other object. The manipulated object remains completely motionless during this attachment phase — no shift, lift, or rotation.
- State which non-manipulated objects remain completely motionless throughout the entire video.
- Only AFTER the strings have visibly attached does the manipulated object begin to move. Describe the subsequent motion as a single smooth arched trajectory:
  - The object rises along a smooth curved path off the surface, reaching a peak clearance of at least 1.5× its own height and at minimum 25 cm near the midpoint of the traversal — clearly airborne with visible empty space beneath it.
  - It moves unidirectionally toward the target along this arc, with no sharp transitions between rise and fall and no flat plateau between them.
  - Throughout the entire motion, the object maintains its elevated clearance from any surface or object beneath it — including the target — and never grazes, drags, scrapes, slides along, or hovers close to anything below it.
  - The object descends from this elevated path only at the final moment the task requires physical contact; if the task does not require contact (i.e., it is performed from above the target), the object holds its elevated position throughout the final action.
  - The final task action is performed in a single smooth motion (e.g., a gentle set-down on contact, or a coordinated single-axis tilt) with no back-and-forth or repeated strokes.
- Emphasize continuous visible frame-by-frame motion through space (no teleportation).

Paragraph 2 — Task completion (∼20–30 words):

- The final accomplished state under a static camera with unchanged backdrop and lighting. The video ends immediately upon task completion.

Return ONLY the 2 paragraphs as flowing prose. No preamble, no headers, no bullet points, no numbered lists in the output.

Figure 3.2: Systematic extended prompt generation template for conditioned video synthesis.

Entity Extraction Protocol

Identify two primary objects from the input prompt for structured interaction analysis.

REQUIRED ENTITIES:

1. **GRASP OBJECT:** The specific item being manipulated via strings.
2. **TARGET OBJECT:** The static object with which the grasp object interacts.

DESCRIPTIVE CONSTRAINTS:

- Use the format: `<basic_color> <object_type>`
- **Color Palette:** Only use red, orange, yellow, green, blue, purple, pink, white, gray, brown, or black.
- **Strict Prohibition:** No materials (e.g., "plastic"), patterns (e.g., "striped"), or shade modifiers (e.g., "light/dark").
- **Precision:** Example: `"blue ladle"` (Correct) vs. `"light blue plastic ladle"` (Incorrect).

INPUT PROMPT: `{extended_prompt_text}`

OUTPUT FORMAT:

```
Grasp: <description>
Target: <description>
```

Figure 3.3: Systematic entity extraction template for object-interaction identification.

While Wan2.2 supports a variety of input modalities, we focus specifically on its image-to-video generation capabilities. We evaluated two variants of Wan2.2: the Text-Image-to-Video model, comprising five billion parameters and supporting 720P resolution at 24 frames per second, and the Image-to-Video model, comprising fourteen billion parameters with support for both 480P and 720P resolutions at 16 frames per second. We adopt the model's default negative prompt, a prompt specifying what the model should avoid generating, and employ FP8 quantization to improve inference efficiency, reducing memory requirements while preserving generation quality, enabling deployment on a single NVIDIA GeForce RTX 4090. From rudimentary tests, generating at a higher resolution 720P did not always improve task adherence, at times worsening it, and was approximately four times slower, taking around three minutes per video.

To provide conditioning frames for the video generation model, we captured fifteen RGBD images per task on a black curtain background using a ZED 2 stereo camera at 480P resolution, aligning with the resolution output of our video model. Across these captures, the position and orientation of the manipulated object are varied while the target object remained stationary, ensuring diversity in the initial conditions of the generated demonstrations for the later policy training. It is worth noting that since the policy operates on relative poses, it is agnostic to which object is moving in the initial images as the same formulation applies equally to cases where the target object moves rather than the grasped object. An example of the video generation from Wan2.2 placing a mug under a coffee machine is illustrated in Figure 3.4.

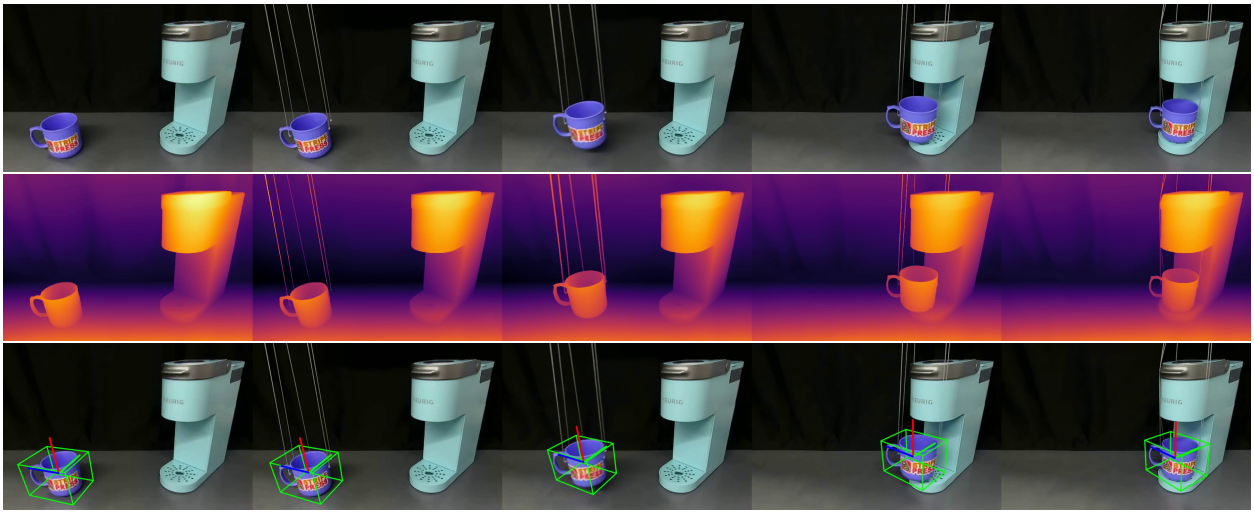

Figure 3.4: An example of the Wan 2.2 generated video, its corresponding generated depth from the extended prompt, and the pose tracking of the grasped object for placing a mug under a coffee machine.

3.2.3 Video Interpolation and Depth Extraction

Since Wan2.2 is trained on 16 fps videos, it generates most accurately at that frame rate. However, 32 fps is needed for reliable object tracking, so we use RIFE [29], a real-time intermediate flow estimation method, to interpolate the generated frames from 16 to 32 fps.

To ground the generated video in real-world spatial structure, we employ Video Depth Anything [30], a video depth estimation model built upon the Depth Anything V2 backbone [31], which has established strong performance in monocular depth estimation. A key advantage of Video Depth Anything over applying frame-level monocular depth models independently is its joint optimization for output quality, temporal consistency, and generalization across diverse scenes, properties that are critical when processing synthetic videos. The interpolated generated output from Wan2.2 is passed through Video Depth Anything, which supports two estimation modes: metric depth and relative depth.

Our system utilizes relative depth estimation, which we found to offer superior robustness across varying scenes compared to metric alternatives. To align these ungrounded relative predictions with the physical environment, we implement a calibration step that solves for a scale and shift transform by comparing the first generated depth frame with the actual depth from the physical camera. The calibration solves for a shift b and scale s in disparity space

via least-squares regression, fitting the linear model

$$\frac{1}{d_{\text{metric}}} = b + s \cdot d_{\text{relative}} \tag{3.1}$$

to all valid pixels in the first frame where both the predicted and reference depth values are non-zero, then applying the inverse transform to recover metric-aligned depth values across the video depth sequence. While metric depth models would be advantageous if they had high fidelity as we could capture 15 RGB images without any depth ground truth, we found that VDA is not sufficiently accurate in this mode.

We additionally experimented with Video Depth Anything Streaming V3 [32], built on a more developed version of Depth Anything that is intended to process long video sequences. However, we found that the model exhibited poor temporal consistency; when tracking the object pose across frames, the result is very jittery, producing non-smooth trajectories that are unsuitable for policy learning. An example of the output from Video Depth Anything is illustrated in Figure 3.4.

3.3 Extracting Object Pose Trajectories

In order to track the object's pose trajectory, we rely on FoundationPose, the current most advanced system for 6D pose estimation and tracking [33]. FoundationPose takes as input the RGBD image observations, camera intrinsic parameters, binary segmentation masks for initial registration, and the mesh of the object we desire to track. As we already have the RGBD observations as detailed in the previous section from our video generation model and the camera intrinsics are known, we need to solve for the object mesh and binary segmentation masks for the target and grasped objects.

3.3.1 Segmentation Masks

Segmentation masks are generated jointly across all object classes in a single inference pass, rather than sequentially per object. Given a scene, text prompts corresponding to all relevant objects — derived from the grasped and target object names extracted in 3.2.1 — are submitted simultaneously to a grounded SAM pipeline. This joint formulation allows non-maximum suppression to operate globally across all detected regions, which is critical for preventing duplicate segmentation in cases where multiple object descriptions could plausibly correspond to the same physical entity. The resulting masks are mapped to their respective object labels and passed to FoundationPose.

Figure 3.5: Example meshes reconstructed from a single image using SAM 3D Objects for the place task. Objects are not shown to scale and have been enlarged for visibility.

3.3.2 Extracting Object Meshes

A fundamental prerequisite for robust object pose tracking is a high-quality 3D mesh of the target rigid body that is both geometrically accurate and metrically consistent with the physical object. We leverage SAM 3D Objects, the latest state-of-the-art model for single-image 3D mesh reconstruction, which produces meshes that are metrically accurate and scaled to real-world dimensions [34]. This allows our system to have a high degree of generality due to the fact that previous systems relied on taking a video or multiple viewpoints of the relevant object to be able to construct the meshes of objects they haven't seen before.

Our pipeline captures an RGB image of the scene and obtains a highly metrically accurate depth map using FoundationStereo which leverages a stereo image pair to produce high-fidelity point cloud reconstructions [35]. The target and grasp objects are then segmented via the method described in the previous subsection, after which depth outliers within each object mask are removed to produce a clean per-object depth point map. The original RGB image, binary segmentation mask, and the filtered depth point map are subsequently passed to SAM 3D Objects, which reconstructs per-object meshes at correct metric scale.

Combining the segmentation masks obtained in Section 3.3.1 with the reconstructed meshes described above, we run FoundationPose to track the 6D object poses throughout each generated video. The accuracy of the resulting pose tracking is illustrated in Figure 3.4, with the corresponding meshes shown in Figure 3.5.

3.4 Policy Training

The policy architecture follows that of SPOT [21] directly. As in SPOT, we adopt the keyframe selection strategy introduced in PerAct [7] to improve training efficiency and avoid learning from continuously dense and potentially noisy pose trajectories. Specifically, a frame is retained only if the relative velocity of the manipulated object is zero, indicating a change in direction, or if its displacement from the most recently retained frame exceeds a predefined translation or rotation threshold. Following keyframe filtering, each pose is augmented with a task completion percentage to enable the policy to determine when the task has been successfully achieved.

3.4.1 Object Trajectory Diffusion

Following the architecture introduced in SPOT and the notation established in Section 3.1.2, the relative pose trajectory $\{\hat{T}^i\}_{i=1}^{l}$ of the grasped object is expressed in the target object's coordinate frame. The model builds upon the Diffusion Policy framework [3], which models the conditional action distribution via a denoising diffusion probabilistic model (DDPM). Crucially, rather than conditioning on raw image observations to predict robot actions, both the conditioning input and the prediction target are object poses expressed in the target object's frame, $\hat{T}_t \in SE(3)$, yielding a compact and embodiment-agnostic formulation.

The object trajectory diffusion model learns to synthesize future poses of the grasped object relative to the target object frame. Formally, we model the conditional distribution $p(\hat{T}_{t+1} \mid \hat{T}_t)$, where $\hat{T}_t \in SE(3)$ denotes the current relative pose of the grasped object with respect to the target, and $\hat{T}_{t+1}$ denotes the next desired pose along the trajectory. Rather than conditioning on raw visual observations, the diffusion process is conditioned solely on the current and historical relative object poses, yielding a compact and embodiment-agnostic representation.

Building upon the DDPM framework, the reverse denoising process iteratively refines a noisy pose according to:

$$\hat{T}_t^{k-1} = \alpha \left(\hat{T}_t^k - \gamma \varepsilon_\theta \left(\hat{T}_t, \hat{T}_{t+1}^k, k \right) + \mathcal{N}(0, \sigma^2 I) \right) \tag{3.2}$$

where ε_θ is the noise prediction network parameterized by θ, k denotes the denoising iteration, and (α, γ, σ) are the parameters of the noise scheduler. The network is trained by minimizing the mean squared error between the predicted and true noise:

$$\mathcal{L} = \text{MSE} \left(\varepsilon_k,\ \varepsilon_\theta \left(\hat{T}_t,\ \hat{T}_{t+1}^0 + \varepsilon_k,\ k \right) \right) \tag{3.3}$$

At inference time, the trained model iteratively denoises from Gaussian noise to produce the next relative pose $\hat{T}_{t+1}$, which is then converted into an end-effector command for closed-loop robot execution. Beyond this core architecture, a lightweight multi-layer perceptron (MLP) pose encoder transforms raw poses into compact latent representations, while a dedicated MLP predicts task progression.

3.4.2 Rejection Sampling

Owing to the stochastic nature of video generation models, a filtering stage is necessary to remove demonstrations that fail to adhere to the conditioning prompt or yield inaccurate pose trajectories. A fundamental challenge inherent to this approach is that generative video models produce outputs probabilistically, meaning that objects may undergo implausible morphological changes or violate physical constraints during generation.

We investigated automated filtering approaches using video understanding models such as VideoScore and NVILA [36][37], but found them insufficiently reliable for this purpose, as evaluating the same video with an identical prompt produced inconsistent scores across runs. Consequently, we currently employ manual rejection filtering, in which generated videos are inspected for physical constraint violations, task completion, and consistency of pose tracking. A promising direction for future work is the adoption of reward models specifically designed for robotic task evaluation, such as RoboReward [38], which could enable scalable and automated quality filtering without the current manual intervention.

3.5 Real-World Rollout

This section describes the real-world execution of our policy and details how each phase of the three-stage pipeline is carried out.

3.5.1 Approach and Grasp

As aforementioned, we use Contact GraspNet to generate grasp proposals on the target object [39], followed by a filtering stage to select viable candidates. Specifically, since our policy is trained to predict task completion percentage, we extract the initial relative pose between the grasped and target object obtained by using the same object meshes and segmentation pipeline described in Section 3.3 and iteratively feed predicted poses into the policy until a completion percentage above 95% is reached.

We approximate the gripper's collision geometry by decomposing the mesh into a set of

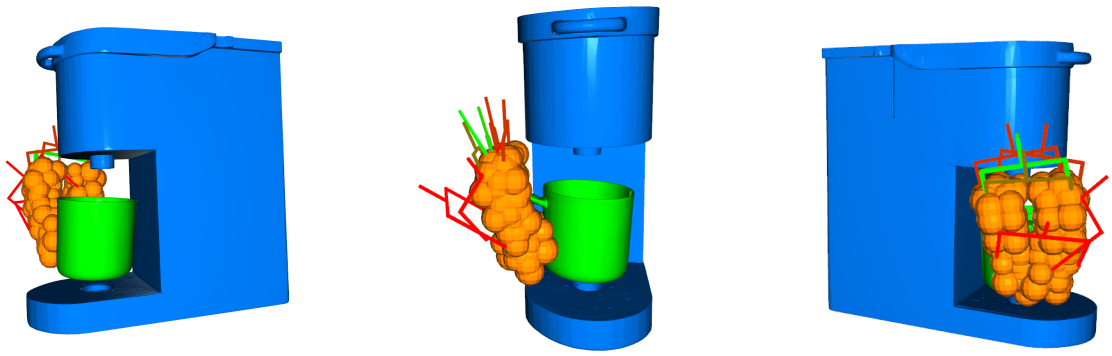

Figure 3.6: Example grasp filtering: grasps along the cup's rim are removed due to potential collision with the top of the coffee machine. Grasp quality is color-coded: green indicates higher stability, red indicates poor stability.

bounding spheres centered at the grasped object's position. Any grasp proposals are discarded if the end-effector trajectory indicates a collision with the target object at the 95% task-completion pose or the four preceding poses, as such collisions would compromise task success. For example, in the pouring task this prevents grasping near the spout, which would collide with the cup. Collision checking is restricted to the final five poses rather than the full trajectory to reduce computational overhead, as collisions in earlier poses are unlikely to affect task success.

Critically, by evaluating collisions directly from the gripper mesh geometry rather than relying on fixed, pre-specified grasp regions like SPOT [21], this approach generalizes naturally to diverse object placements and orientations without requiring any task-specific heuristics. After filtering, we select the grasp proposal with the highest confidence score from Contact GraspNet among the remaining candidates. An example is shown in Figure 3.6 in which the grasps along the rim of the cup have been filtered out as a result of the fact that they would collide with the top of the coffee machine.

3.5.2 Object-Manipulation

During the object manipulation phase, only the grasped object's pose is tracked continuously, as we only require the target object's starting pose as it remains stationary throughout execution. At each timestep, the trajectory diffusion model predicts a horizon of four future poses expressed relative to the target object frame. Following a receding horizon control strategy, only the first predicted pose is executed; the relative transformation between the current and next predicted pose is computed in SE(3) and mapped directly onto the robot's end-effector frame, after which inverse kinematics is solved to determine the corresponding

joint configuration. Once executed, the updated real-time pose is fed back into the policy, forming a closed-loop control scheme that accounts for accumulated errors and unexpected object movements during execution.

3.5.3 Retract and Reset

Similar to the approach and grasp phase, we denote the task as being successfully completed once the completion percentage is predicted to be above 95% and our resetting actions are defined at this phase of the policy: (1) releasing the object or (2) returning the grasped object to its initial position. Regardless of which action is taken, the robot ultimately returns to its initial pre-task position, effectively resetting the system for another execution of the task.

3.6 Characteristics of an Effective Policy

Our proposed methodology successfully satisfies the four requirements outlined in Section 1.2 as follows.

1. **Data Collection Efficiency:** Data collection is substantially more efficient than conventional approaches. Rather than requiring teleoperated or human demonstrations, our pipeline necessitates only the capture of fifteen images in which the grasp object is repositioned between shots. Demonstration generation is then fully automated via the video generation model, eliminating the need for manual intervention.

2. **Sample Efficiency:** By virtue of the compact and informative nature of object-centric pose trajectories, our policies can be learned from as few as ten to fifteen trajectories after filtering, in contrast to the hundreds of demonstrations typically required by end-to-end visuomotor approaches.

3. **Meaningful Environment Representation:** Our framework operates directly on extracted object poses without relying on intermediate latent representations. This allows the policy to efficiently encapsulate task-relevant dynamics in their most essential form, learning directly from the input representation without the overhead of encoding raw visual observations.

4. **Generality:** By operating on object pose trajectories, the system generalizes naturally across varying object positions, appearances, and environmental conditions, remaining robust to visual distractors, lighting variation, and scene changes. Critically, the embodiment-agnostic nature of the object-centric representation enables seamless transfer across different robotic platforms without retraining.

Chapter 4

Results

This chapter presents the experimental results evaluating the effectiveness and generalizability of the proposed methodology, followed by a discussion of its limitations.

4.1 Tasks

4.1.1 Experimental Tasks

The following tasks were evaluated, each accompanied by a natural language prompt as shown in Table 4.1. An illustrative example of a corresponding extended prompt, from which the grasp and target object names are extracted, is provided in Figure 4.1. The video generations of all tasks are displayed in Figure 4.2.

Table 4.1: Task descriptions and associated objects used in real-world evaluation.

Task	**Prompt Description**	**Grasp Object**	**Target Object**
Insert	Insert the book in the shelf	Green Book	Red Shelf
Stack	Stack the yellow bowl inside the blue bowl	Yellow Bowl	Blue Bowl
Align	Align the green mug on top of the blue mug	Green Mug	Blue Mug
Pour	Pour water into the teacup	Blue Pitcher	Orange Tea Cup
Place	Place the mug under the coffee machine	Purple Mug	Blue Coffee Maker
Hang	Hang the clothes hanger onto the rack	Blue Hanger	White Rack

Input / Output

INPUT:

Place the mug under the coffee machine

OUTPUT:

No hands, no human figures, no body parts appear at any time. The video opens with the scene perfectly static. Four thin semi-transparent strings visibly descend from the top of the frame and attach to the mug's rim and handle. The mug remains motionless during this attachment phase, while the coffee machine and background remain perfectly static throughout the entire video. Only after the strings have visibly attached does the mug rise along a smooth arched trajectory, reaching a peak clearance of 30 centimeters above the surface with clear empty space beneath it. It moves unidirectionally toward the coffee machine with continuous, frame-by-frame motion, never grazing any surface or the machine itself. After traversing the arc, the mug descends vertically under the dispenser. The strings remain taut and visible as the mug is placed with a single, gentle set-down. The mug retains its exact shape and color throughout the movement.

The mug sits centered under the coffee machine's nozzle. The lighting and background remain unchanged as the video ends immediately upon the mug reaching the base.

Figure 4.1: An example input and its corresponding output prompt, generated following the instructions in Figure 3.2.

4.2 Real-World Results

4.2.1 Robot Embodiment Results

This section reports success rates across two different robot embodiments. To validate the embodiment-agnostic nature of our policy, we evaluated it on two platforms: the Franka Emika Panda (7-DOF) and the UR5 (6-DOF). The same pipeline was deployed across both platforms; the only modifications made were to the grasp execution offsets and the inverse kinematics solver, both of which were strictly necessitated by the inherent hardware differences between the two robots. All other experimental conditions, including objects and models, were identical with one another. Success rates for each platform are reported in Table 4.2 and the majority of failures are attributed to pose tracking loss. An example of the task of putting the mug under the coffee machine on both embodiments is illustrated in Figure 4.3.

We are currently working on a comprehensive comparison against several baselines, including the original SPOT policy trained on human demonstrations and diffusion policy to demonstrate that our approach can achieve competitive performance using exclusively synthetic demonstration data.

Figure 4.2: Representative frames sampled at five equal intervals across each of the six demonstration tasks.

4.2.2 Runtime Analysis

Since all data generation is performed on a single NVIDIA GeForce RTX 4090, with the exception of mesh generation which is performed by a NVIDIA GeForce RTX 3090, the central questions are the effectiveness of our proposed methodology in autonomous data generation and the data collection efficiency of policy learning. The breakdown in the time it takes for each process is illustrated in Table 4.3 which provides a high level overview of the key steps involved in deriving the policy. As illustrated in the table, the entire policy is learnable in just over one hour, underscoring the computational efficiency of our autonomous learning framework. Further optimizations are possible, as current latency is partially attributed to the generation of visualizations for quality verification.

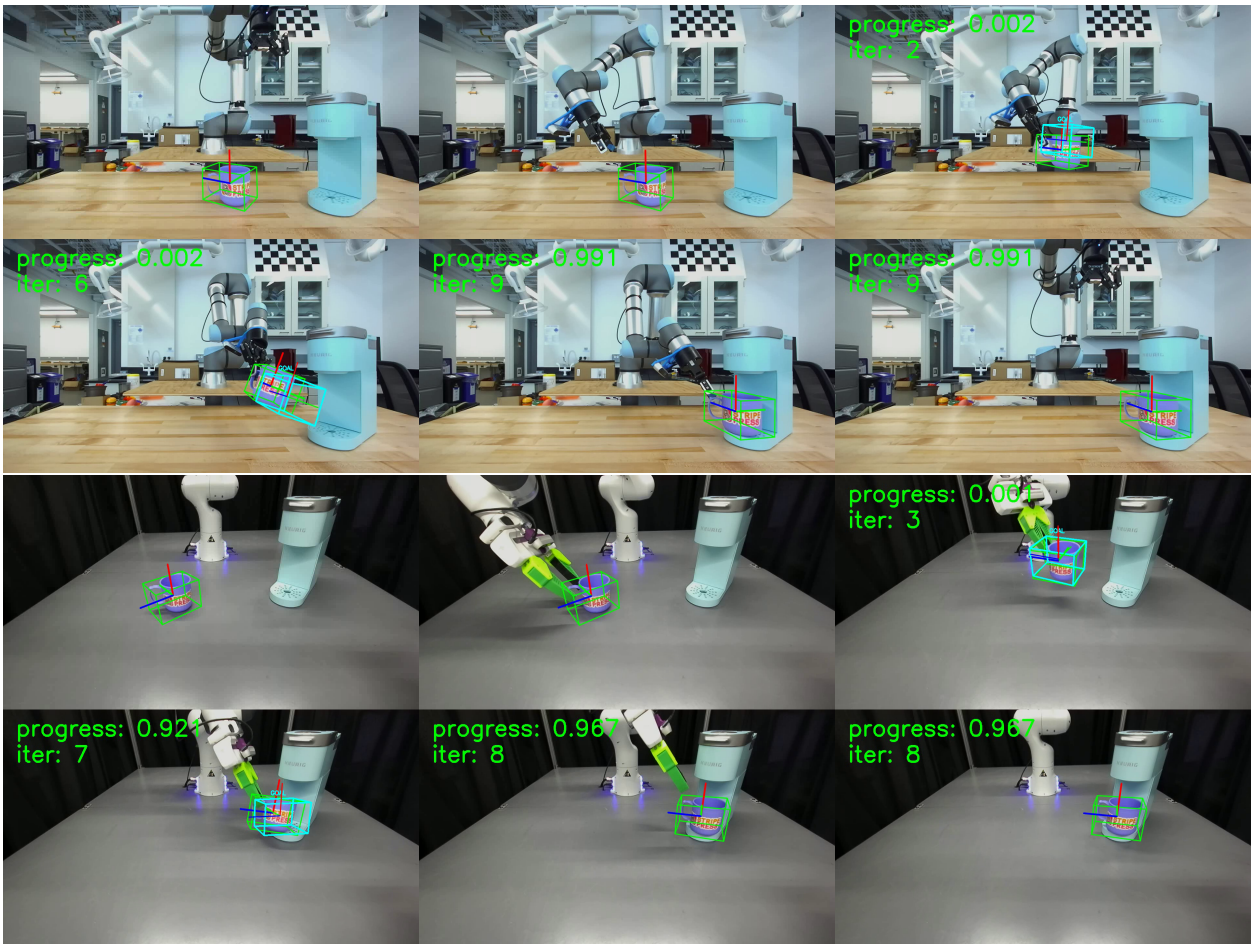

Figure 4.3: Cross-robot rollout demonstrations on the Panda and UR5 using a shared policy, with the progress prediction bar shown in the upper left.

4.3 Limitations

4.3.1 Symmetric Objects

A limitation of the current system is that tasks require non-symmetric objects. For instance, if we were to take two completely solid colored bowls and the task at hand was to stack one on top of the other, it becomes difficult to determine a consistent canonical pose across the demonstrations because both objects are symmetric. Furthermore, since the mesh is reconstructed from a single image, SAM 3D Objects is inherently limited to observing only the front surface of the object. Consequently, the model tends to extrapolate the unobserved geometry by mirroring the visible front features, potentially producing a reconstruction that exhibits unintended symmetry.

Table 4.2: Success rates across tasks for Franka Panda and UR5.

Robot	**Insert**	**Stack**	**Align**	**Pour**	**Place**	**Hang**
Franka Panda	4/5	5/5	4/5	4/5	4/5	3/5
UR5	3/5	5/5	3/5	4/5	5/5	3/5

Table 4.3: Runtime breakdown of the proposed methodology pipeline.

Step	**Runs**	**Time per Run**	**Total Time**
Data Collection			
Image Collection	1	1 min	1 min
Mesh Extraction	2	2 min	4 min
Video Generation	15	47 s	∼12 min
Data Processing			
Interpolation	15	4 s	1 min
Depth Prediction	15	18 s	∼5 min
Masking	30	18 s	9 min
Pose Tracking	30	26 s	∼13 min
Policy Learning			
Policy Training	1	25 min	25 min
Total			∼70 min

4.3.2 Video Generation Capabilities

Additionally, the video generation model exhibited difficulty with tasks outside its training distribution. For instance, it failed to generate plausible demonstrations of placing a mug on a mug tree, consistently hanging the handle below the hook rather than over it, even when conditioned on human hand demonstrations. This suggests that open-source video generation models still lack grounded physical reasoning and that significant progress remains to be made in this area.

Furthermore, the video generation instructions were determined empirically through iterative refinement guided by observed failure modes. Specifically, we constrained the model to maintain the final object configuration upon task completion; without this specification, the fixed five-second video duration would result in the generation of unintended continuations beyond the goal state. Therefore, one area of exploration is whether the more powerful paid models are capable of generating more complex demonstrations with less human guidance as for instance NovaFlow and RIGVID both depend on proprietary models [25][24].

4.3.3 Occlusion

A key limitation of tasks that rely on an external camera is occlusion, which may arise either from the selected grasp during execution or from the robot moving into configurations that obstruct the camera's view of relevant objects midway through the execution. Because the system always involves manipulating one object, the act of grasping inherently introduces partial occlusion, which can cause the pose tracker to lose track of the object. Although our system currently uses a single external camera, because the policy is object-centric, the underlying premise is that the specific viewpoint is not critical as long as both objects remain visible. This suggests that the system's robustness could be further improved in multi-camera settings: as long as the object pose is observable and trackable from at least one camera, the policy can still be successfully executed.

Chapter 5

Conclusion

This final chapter outlines the promising directions for future research, highlighting the broader potential of the framework developed throughout this thesis.

5.1 Future Work

5.1.1 Object Generalization

While the proposed policy framework demonstrates strong generalization across robot embodiments and visual backgrounds, generalization to novel object instances within the same category remains a limitation. Consider the bowl stacking task: substituting the original bowls with a new set is feasible provided that the canonical poses of the replacement objects are consistent with those used during training. Bowl stacking is a relatively forgiving case in this regard, as successful execution depends primarily on positional accuracy and is largely invariant to the object's rotation about the vertical axis.

However, tasks where orientation is more significant, such as stacking cups with handle alignment, are considerably more sensitive to object-level variation, as small differences in canonical pose or object geometry can directly cause task constraint violations. This motivates the need for a method of aligning the canonical pose of a novel object instance to that of the training object, such that the learned trajectory can still roll out successfully.

An additional axis of generalization comes from the policy's implicit encoding of object-specific geometry in the learned trajectory. In the pouring task, for instance, if the policy has learned to position the spout near the rim of a particular cup size, reducing the cup's diameter while preserving the same canonical pose would cause the learned trajectory to miss the target entirely. This suggests that future work should explore conditioning the policy on

object shape or scale parameters to enable more robust generalization across intra-category geometric variation.

5.1.2 Modularity of Components

A notable property of the proposed system is its modularity: each component of the pipeline interfaces with the learned policy through well-defined inputs and outputs, such that individual modules can be replaced with improved alternatives as the underlying foundation models advance that can offer gains in quality or efficiency. For example, the current open-source video generation model could be substituted with a more capable proprietary model offering stronger physics understanding and prompt adherence, potentially alleviating the failure modes. Similarly, advances in monocular depth estimation that enable accurate metric depth prediction would remove the need for dedicated depth capture and scale recovery, allowing data collection to be performed with generic RGB cameras and further reducing the requirements of the system.

5.1.3 Learning a Policy from a Single Image

Currently, our data collection requires fifteen RGBD images to generate the initial frames used to prompt the image to video model. An interesting avenue for future exploration, however, is whether a single RGBD capture could be sufficient, with the remaining images synthesized automatically.

One promising direction is to remove the objects from a single RGBD capture, recover the clean background via inpainting, and re-render our extracted meshes back into the scene across a diverse range of poses. This would enable the synthesis of an arbitrary number of RGBD training samples from a single observation, potentially exceeding the fifteen images currently required. A key challenge, however, is the synthetic domain gap introduced by re-rendering as it remains unclear whether video generation models trained on real imagery can generalize reliably when conditioned on rendered inputs.

A complementary approach would leverage recent advances in structured, object-centric scene editing. Neural USD introduces a hierarchical scene representation that enables precise, per-object control over appearance, geometry, and pose [40]. Such a framework could address the domain gap concern by generating more photorealistic and diverse initial frames from a single capture. Therefore, it may be possible that a manipulation policy can be learned from a single RGBD capture.

5.1.4 Automated Rejection Filtering

In conjunction with the single-image policy learning framework described previously, the current reliance on manual filtering of implausible demonstrations could be replaced by automated heuristics operating directly on the extracted pose trajectories, or by leveraging task completion models such as RoboReward [38]. Beyond filtering, such models could additionally provide dense reward signals over the course of a trajectory, enabling the system to retain partially successful demonstrations and weight them appropriately during training, rather than discarding them entirely.

5.2 Concluding Remarks

This thesis set out to address one of the core challenges in robotic manipulation: the ability to learn effective, generalizable policies efficiently. Through the combination of single image mesh extraction, video generation models, object-centric policy learning, and our three-stage task decomposition, we have demonstrated that capable manipulation policies can be acquired autonomously from only a text prompt and a small number of images, requiring no human or simulated demonstrations whatsoever.

With the continuous improvement of foundation models, every component of the pipeline stands to benefit. More capable video generation models will yield more physically plausible trajectories, stronger depth estimation will simplify data collection, and more accurate pose tracking will improve execution. The modularity of the framework means these advances translate directly into performance gains without any redesign.

This thesis has taken steps toward addressing the longstanding challenge of building capable and autonomous robotic manipulation systems, presenting generative models as a promising and scalable alternative to conventional demonstration collection. Taken together, the contributions of this work point toward a broader vision in which a robot need only observe a single image of a scene and receive a text prompt to autonomously synthesize its own training data and acquire a task-specific policy, with no human intervention required.

References

[1] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. *Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots*. 2024. arXiv: 2402.10329 [cs.RO].

[2] A. Khazatsky et al. "DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset." *RSS*, 2024.

[3] 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*. 2024. arXiv: 2303.04137 [cs.RO].

[4] A. Brohan et al. *RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control*. 2023. arXiv: 2307.15818 [cs.RO].

[5] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. *Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware*. 2023. arXiv: 2304.13705 [cs.RO].

[6] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. "3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations." In: *Proceedings of Robotics: Science and Systems (RSS)*. 2024.

[7] M. Shridhar, L. Manuelli, and D. Fox. "Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation." In: *Proceedings of the 6th Conference on Robot Learning (CoRL)*. 2022.

[8] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. *RVT: Robotic View Transformer for 3D Object Manipulation*. 2023. arXiv: 2306.14896 [cs.RO].

[9] R. Firoozi et al. *Foundation Models in Robotics: Applications, Challenges, and the Future*. 2023. arXiv: 2312.07843 [cs.RO].

[10] A. Agarwal, G. Singh, B. Sen, T. Lozano-Pérez, and L. P. Kaelbling. *SceneComplete: Open-World 3D Scene Completion in Cluttered Real World Environments for Robot Manipulation*. 2025. arXiv: 2410.23643 [cs.RO].

[11] W. Shen, N. Kumar, S. Chintalapudi, J. Wang, C. Watson, E. S. Hu, J. Cao, D. Jayaraman, L. P. Kaelbling, and T. Lozano-Pérez. *TiPToP: A Modular Open-Vocabulary Planning System for Robotic Manipulation*. 2026. arXiv: 2603.09971 [cs.RO].

[12] A. Chapin, B. Machado, E. Dellandrea, and L. Chen. *Object-Centric Representations Improve Policy Generalization in Robot Manipulation*. 2025. arXiv: 2505.11563 [cs.RO].

[13] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. *Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation*. 2021. arXiv: 2112.05124 [cs.RO].

[14] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki. "SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion." *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[15] T. Weng, D. Held, F. Meier, and M. Mukadam. *Neural Grasp Distance Fields for Robot Manipulation*. 2023. arXiv: 2211.02647 [cs.RO].

[16] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. "Flow as the Cross-domain Manipulation Interface." In: *Conference on Robot Learning*. PMLR. 2025, pp. 2475–2499.

[17] C. Yuan, C. Wen, T. Zhang, and Y. Gao. *General Flow as Foundation Affordance for Scalable Robot Learning*. 2024. arXiv: 2401.11439 [cs.RO].

[18] X. Fang*, B.-R. Huang*, J. Mao*, J. Shone, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling. "KALM: Keypoint Abstraction using Large Models for Object-Relative Imitation Learning." In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2025.

[19] J. Gao, Z. Tao, N. Jaquier, and T. Asfour. "K-VIL: Keypoints-Based Visual Imitation Learning." *IEEE Transactions on Robotics*, **39**(5), Oct. 2023, pp. 3888–3908. ISSN: 1941-0468. DOI: 10.1109/tro.2023.3286074.

[20] B. Wen, W. Lian, K. Bekris, and S. Schaal. *You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration*. 2022. arXiv: 2201.12716 [cs.RO].

[21] C.-C. Hsu, B. Wen, J. Xu, Y. Narang, X. Wang, Y. Zhu, J. Biswas, and S. Birchfield. *SPOT: SE(3) Pose Trajectory Diffusion for Object-Centric Manipulation*. 2025. arXiv: 2411.00965 [cs.RO].

[22] P. Vitiello, K. Dreczkowski, and E. Johns. *One-Shot Imitation Learning: A Pose Estimation Perspective*. 2023. arXiv: 2310.12077 [cs.RO].

[23] S. Zhou, Y. Du, J. Chen, Y. Li, D.-Y. Yeung, and C. Gan. *RoboDreamer: Learning Compositional World Models for Robot Imagination*. 2024. arXiv: 2404.12377 [cs.RO].

[24] S. Patel, S. Mohan, H. Mai, U. Jain, S. Lazebnik, and Y. Li. "Robotic Manipulation by Imitating Generated Videos Without Physical Demonstrations." In: *The Fourteenth International Conference on Learning Representations*. 2026.

[25] H. Li, L. Sun, Y. Hu, D. Ta, J. Barry, G. Konidaris, and J. Fu. *NovaFlow: Zero-Shot Manipulation via Actionable Flow from Generated Videos*. 2025. arXiv: 2510.08568 [cs.RO].

[26] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal. *Compositional Foundation Models for Hierarchical Planning*. 2023. arXiv: 2309.08587 [cs.LG].

[27] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. *Learning to Act from Actionless Videos through Dense Correspondences*. 2023. arXiv: 2310.08576 [cs.RO].

[28] T. Wan et al. *Wan: Open and Advanced Large-Scale Video Generative Models*. 2025.

[29] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou. "Real-Time Intermediate Flow Estimation for Video Frame Interpolation." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022.

[30] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang. *Video Depth Anything: Consistent Depth Estimation for Super-Long Videos*. 2025. arXiv: 2501.12375 [cs.CV].

[31] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. *Depth Anything V2*. 2024. arXiv: 2406.09414 [cs.CV].

[32] H. Lin, S. Chen, J. H. Liew, D. Y. Chen, Z. Li, G. Shi, J. Feng, and B. Kang. *Depth Anything 3: Recovering the visual space from any views*. 2025. arXiv: 2511.10647 [cs.CV].

[33] B. Wen, W. Yang, J. Kautz, and S. Birchfield. *FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects*. 2024. arXiv: 2312.08344 [cs.CV].

[34] S. 3. Team et al. *SAM 3D: 3Dfy Anything in Images*. 2025. arXiv: 2511.16624 [cs.CV].

[35] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield. *FoundationStereo: Zero-Shot Stereo Matching*. 2025. arXiv: 2501.09898 [cs.CV].

[36] X. He et al. *VideoScore: Building Automatic Metrics to Simulate Fine-grained Human Feedback for Video Generation*. 2024. arXiv: 2406.15252.

[37] Z. Liu et al. *NVILA: Efficient Frontier Visual Language Models*. 2024. arXiv: 2412.04468 [cs.CV].

[38] T. Lee, A. Wagenmaker, K. Pertsch, P. Liang, S. Levine, and C. Finn. *RoboReward: General-Purpose Vision-Language Reward Models for Robotics*. 2026. arXiv: 2601.00675 [cs.RO].

[39] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. *Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes*. 2021. arXiv: 2103.14127 [cs.RO].

[40] A. Escontrela, S. Kushagra, S. v. Steenkiste, Y. Rubanova, A. Holynski, K. Allen, K. Murphy, and T. Kipf. *Neural USD: Scalable Scene Editing via Differentiable Universal Scene Description*. 2025. arXiv: 2510.23956 [cs.CV].