PROXY: Learning Object-Centric Manipulation Policies from a Single RGB-D Image

Maggie Huili Yao, Sahit Chintalapudi, Akash Anand, Megha Tippur,
Tomás Lozano-Pérez, Leslie Pack Kaelbling, Aditya Agarwal

***Abstract*— Robot manipulation policies conventionally rely on tens to hundreds of teleoperated or kinesthetically guided demonstrations, a manual burden that must be repeated for each new task, environment, and robot embodiment. Recent methods mitigate this by learning robot-agnostic, object-centric policies from human demonstration videos, yet collecting such videos still require manual effort and captures a limited distribution of object configurations. We argue that video generative models can serve as a scalable replacement for human data collection, autonomously synthesizing diverse demonstrations across object placements at zero human cost. We introduce PROXY, a method that learns a task-specific policy from a single RGB-D image and a natural-language task description without any real-world data collection. PROXY autonomously reconstructs task-relevant object meshes, diversifies scene configurations, synthesizes demonstration videos via a video generative model, and filters implausible video samples through a multi-axis VLM-based verification stage, yielding training data for an object-centric diffusion policy over the relative pose trajectory between a held object and a fixture object. On four real-robot tasks across two embodiments, PROXY matches policies trained on an equivalent number of human demonstrations and outperforms a strong visuomotor baseline trained on real teleoperation data. Critically, broadening the scene-diversification range incurs no additional human cost yet yields out-of-distribution generalization that real-data baselines fail to achieve.**

I. Introduction

Imitation learning has become a leading approach to training robot manipulation policies, yet each new task or environment continues to require dedicated manual effort to collect demonstrations. Behavior-cloning policies [1], [2] typically require tens to hundreds of demonstrations per task, gathered through kinesthetically guided teleoperation [3], handheld manipulators [4], and exoskeletons [5], [6]. Crucially, none of these interfaces eliminates the need for human involvement when tasks, environments, or robot embodiments change; the data-collection effort must simply be repeated from scratch.

Several lines of work attempt to reduce this data-collection effort. Object-centric methods [7], [8], [9] represent scenes via low-dimensional intermediate representations such as keypoints, object poses, and affordances, which improves generalization to viewpoint and lighting changes, and typically requires fewer demonstrations than policies trained on image data, but still require many demonstrations to train the policy. Video-based methods sidestep the need for real-world demonstrations by generating robot actions from pretrained video generative models. They either learn an inverse-dynamics model to infer actions directly from the generated frames [10], [11], which requires explicit action supervision and transfers unreliably to a physical robot, or distill the generated video into an intermediate actionable representation, such as 3D flow, keypoints, or object poses, that is retargeted to the robot [12], [13], [14], [15], [16]. In either case, video generation remains in the deployment-time loop and a fresh video must be synthesized for every object configuration, often taking minutes per generation, so these methods do not generalize to new object configurations or environments without regenerating.

Rather than generating a video for each new scene, we use a video generative model at training time to synthesize a diverse set of training demonstrations at zero human cost, and learn a single policy from them. Concretely, given only a single RGB-D image and a task description, PROXY operates as a fully automated pipeline with no human demonstrations or manual intervention. It reconstructs task-relevant object meshes, generates diverse physically plausible scene configurations, synthesizes a demonstration video for each configuration using a pretrained video generative model, and filters out implausible samples with a multi-axis VLM. From the accepted videos, it extracts the SE(3) pose trajectory of the held object relative to a fixture object and trains an object-centric diffusion policy on these trajectories. Crucially, because the synthetic demonstration distribution is under explicit programmatic control, PROXY generalizes to broader deployment conditions simply by expanding the scene diversification range, a capability that methods reliant on recorded demonstrations cannot replicate without collecting additional data.

The closest predecessor to our approach is SPOT [9], which trains a diffusion policy over SE(3) object pose trajectories, reducing data collection to human demonstration videos rather than robot demonstrations. PROXY takes a similar approach but takes this one step further: by replacing human demonstration videos with automatically generated synthetic ones, it eliminates the remaining manual effort entirely, requiring only a single RGB-D image and a task description.

Our key contributions are as follows: (1) An **automated pipeline requiring no human intervention**: given only a single RGB-D image and a short task description, generates physically grounded synthetic training trajectories with no teleoperated or human demonstrations; (2) **Full-trajectory**

All authors are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Correspondence: {yaohuili, adityaag}@mit.edu

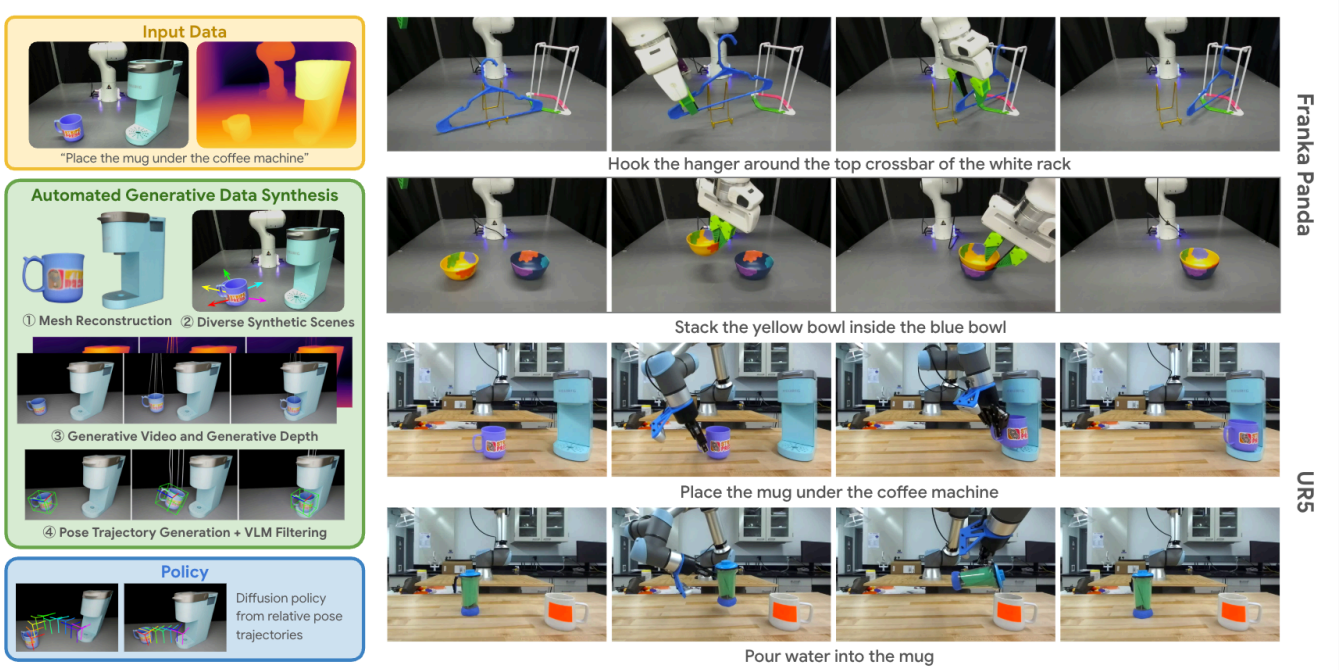


Fig. 1. PROXY learns object-centric manipulation policies from a single RGB-D image and a short task description, with zero real-world demonstrations. We show four real-robot rollouts across two embodiments, a Franka Panda and a UR5, with two tasks per robot.

kinematic feasibility filter: grasp proposals are filtered by forward-simulating the policy to predict end-effector trajectories, rejecting candidates with gripper-mesh collisions or infeasible inverse kinematics, enabling generalization to novel object placements without task-specific heuristics; (3) **Zero-cost distribution broadening**: since scene diversification is synthetically controlled, PROXY generalizes to broader deployment conditions by simply expanding the data generation range – requiring no additional human effort, unlike methods trained on recorded demonstrations.

On four real-robot manipulation tasks on two robot platforms, PROXY **matches** a method sharing the same object-centric relative pose policy formulation but trained on real human-hand demonstrations, **outperforms** diffusion policy [1] trained on real teleoperation data, and recovers out-of-distribution performance that real-data baselines fail to achieve.

II. Related Work

a) Scalable Data Generation for Robot Learning: The dominant paradigm for training manipulation policies is to collect demonstration data directly on the robot through teleoperation and learning visuomotor policies from them [2], [1], [17], [3]. Since teleoperating the robot for every task is costly, cheaper interfaces such as handheld manipulators [4], [18] and wearable exoskeletons [5], [19] lower this effort, but a human must still physically perform each demonstration. To avoid this per-task human effort altogether, a complementary line of work synthetically generates demonstrations, using simulation to amplify a small set of real demonstrations into many [20], [21], [22], or to procedurally construct diverse scenes and tasks [23], [24], [25], though still relying on seed demonstrations and hand-build simulation assets. PROXY instead generates training demonstrations directly from a single real RGB-D image and a task description, requiring neither seed demonstrations nor simulation engineering.

b) Video Generative Models for Robot Learning: A growing body of work uses pretrained video generative models for generating robot actions for manipulation. Some methods [10], [11] infer actions directly from the generated frames with an inverse-dynamics model, which requires action-labeled supervision and transfers unreliably to a physical robot. Another line of methods distill the generated video into an intermediate actionable representation: 3D object flow, keypoints, or object poses, that is retargeted to the robot [12], [13], [14], [15], [16], [26]. While these methods avoid collecting real demonstrations, they keep video generation in the deployment loop, and a fresh video must be generated, and the actions re-extracted for every new object configuration, which is slow and fails to generalize. Rather than generating video at deployment, PROXY generates videos offline to build a training set and distills them into a reusable closed-loop object-centric policy. At deployment time, the policy generalizes to new object placements, without per-instance regeneration.

c) Object-Centric and Structured-Intermediate Policy Learning: Object-centric and structured-intermediate representations make manipulation policies more sample-efficient and more robust to viewpoint, pose, and lighting variation. These methods differ in what they ultimately output. One group learns a policy that maps the intermediate represen-

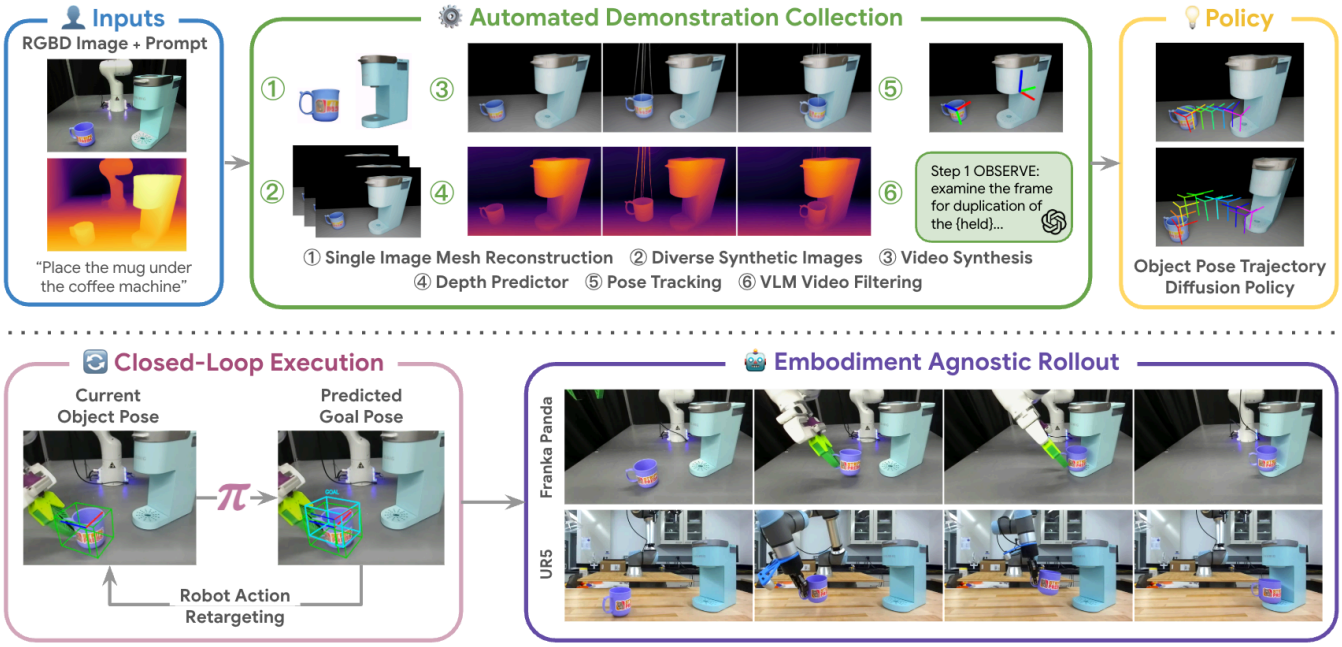


Fig. 2. Given a single RGB-D image and a natural-language task description, PROXY reconstructs task-relevant 3D object meshes and generates a diverse set of physically plausible scene configurations. For each configuration, it synthesizes a demonstration video using a pretrained video diffusion model, estimates per-frame depth, and applies VLM-based rejection filtering to remove physically implausible samples. Object poses are tracked across frames and used to train an object-centric SE(3) pose-trajectory diffusion policy, which is deployed on real robots via closed-loop execution with action retargeting across embodiments.

tation – dense or task-specific keypoints [7], [27], [28], [29], implicit neural object fields [8], [30], or task-relevant affordances [31] – to end-effector commands. A second group predicts an object-centric quantity rather than robot actions. Pose-trajectory methods such as SPOT [9] predict a trajectory of relative SE(3) poses between two task-relevant objects, which is converted to robot motion at execution time. PROXY follows this second formulation, predicting relative object pose trajectories rather than embodiment-specific commands, which makes the learned policy transferable across robots.

d) Closest Work: The most direct predecessor to our work is SPOT [9], which introduces an $\mathrm{SE}(3)$ pose-trajectory diffusion policy operating between a held and fixture object. By virtue of its object-centric representation, SPOT can be trained from human-hand demonstration videos rather than on-robot data. However, three limitations remain. First, a human must still record demonstration videos for each task. Second, SPOT recovers object meshes via BundleSDF [32] applied to the demonstration video, coupling mesh quality to the multi-view coverage incidentally provided by the demonstrator. Third, grasp poses are copied directly from the demonstrations, preventing generalization to novel object placements at deployment time. Our work addresses each limitation: synthetic videos generated by a video generative model [33] and filtered through a multi-axis VLM-based rubric replaces human demonstrations completely; single-image mesh recovery [34] replaces BundleSDF and decouples mesh extraction from demonstration collection; and grasp proposals generated by an off-the-shelf model [35] are screened by our full-trajectory kinematic feasibility filter, replacing grasps copied from demonstration videos and generalizing to novel object placements.

III. Method

A. Problem Formulation and Overview

Given a single RGB-D image $\mathbf{I}_{\text{real}}$ of a tabletop scene and a short natural-language task description ℓ, we train a manipulation policy π entirely from synthetically generated demonstrations. We consider single-arm manipulation tasks involving two task-relevant objects: a held object o_h (the object to be grasped and manipulated) and a fixture object o_f (the stationary reference object). Let $T_{h,t}, T_{f,t} \in \mathrm{SE}(3)$ denote the poses of o_h and o_f at frame t, and $\tau = (T_t^{\text{rel}})_{t=1}^{T}$ the relative pose trajectory, where $T_t^{\text{rel}} = T_{f,t}^{-1} T_{h,t}$. By expressing the trajectory of o_h in the frame of o_f, τ encodes the task-relevant motion in a canonical object-centric space that is invariant to absolute scene configuration.

Our pipeline has three stages: (i) *3D mesh reconstruction and diverse scene generation* (Section III-B): reconstructs task-relevant 3D meshes and estimates their poses from $\mathbf{I}_{\text{real}}$, and generates diverse physically plausible scene initializations rendered as conditioning images for video synthesis; (ii) *Generating synthetic demonstrations* (Section III-C): synthesizes and filters video demonstrations for each scene initialization using an image-and-text-conditioned video model, then extracts metric-aligned depth and SE(3) relative pose trajectories using the reconstructed meshes; and (iii) *Policy training and inference* (Section III-D): trains an object-centric SE(3) pose-trajectory diffusion policy on the

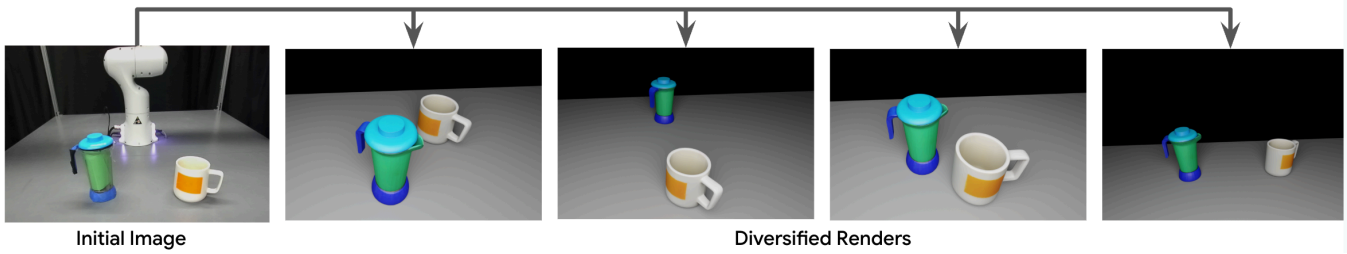


Fig. 3. Scene diversification. Left: the original RGB image $\mathbf{I}_{\text{real}}$ of the *Pour* task. Right: diversified scene initializations $\{\mathbf{I}_0^{(j)}\}$ produced by sampling planar translations and z-axis rotations of the recovered object meshes, settling the perturbed state in PyBullet against the recovered floor mesh, and rendering each settled state from matched camera parameters in Blender.

extracted trajectories, and selects collision-free and kinematically feasible grasps before rolling out the policy on the robot in a closed-loop manner.

B. 3D Mesh Reconstruction and Diverse Scene Generation

a) 3D Mesh Reconstruction: We reconstruct metrically aligned 3D meshes $\mathcal{M}_h, \mathcal{M}_f$ of o_h and o_f from $\mathbf{I}_{\text{real}}$ to enable physically plausible scene rendering and to serve as inputs for pose tracking. We query Gemini 2.5 Flash [36] with $\mathbf{I}_{\text{real}}$ and ℓ to identify the two objects, pass these to GroundedSAM [37] for segmentation, and reconstruct per-object meshes via SAM3D [34]. Finally, FoundationPose [38] is run on each mesh against the metric depth map to obtain the metrically aligned initial pose $T_{i,0}$ of each object in the camera frame.

b) Diverse Scene Generation: We construct a PyBullet [39] simulation with a planar table, placing $\mathcal{M}_h$ and $\mathcal{M}_f$ at their estimated poses $T_{h,0}, T_{f,0}$ with camera extrinsics matched to $\mathbf{I}_{\text{real}}$. We diversify the scene by applying collision-checked planar translations of up to $0.05\,\text{m}$ and rotations of up to 20° about the z-axis to each object, producing 25 physically plausible initializations. These are rendered in Blender [40] to produce conditioning images for video generation, as shown in Fig. 3; the redundancy ensures a sufficient number of valid demonstrations survive the subsequent filtering stage.

C. Generating Synthetic Demonstrations

We train our policy on relative $\text{SE}(3)$ pose trajectories extracted from synthetically generated demonstration videos. For each of the 25 scene initializations, we generate candidate videos using an image-and-text-conditioned video generation model, filter them for quality, and extract metric aligned $\text{SE}(3)$ pose trajectories from the filtered videos, targeting a final set of 12 demonstrations.

a) Video Generation and Filtering: We use Wan 2.2 [33], a state-of-the-art open-source video generation model, to generate demonstration videos conditioned on the 25 synthetically rendered scene images from the previous step as demonstrated in Fig. 2. We generate videos at 480p resolution and 16 fps; increasing to 720p does not improve task adherence and takes approximately four times longer. We use RIFE [41] to interpolate each video to 32 fps for reliable pose tracking.

Wan 2.2 requires both a descriptive prompt and a negative prompt. To minimize hallucinations, we use Gemini 2.5 Flash [36] to elaborate ℓ and $\mathbf{I}_{\text{real}}$ into a detailed prompt ℓ^*, which is reused across all demonstrations for a given task. The full prompt template is provided in the appendix.

As the rendered image contains no robot or human, asking the video model to move the object along a trajectory requests for a motion with no cause. As a result, the generated videos tend to be physically implausible, with objects teleporting or floating – a consequence of the missing agent, not a model deficiency. We therefore add a causal agent: a human hand could provide this cue but introduces occlusions that corrupts depth estimation. Instead, we use a suspended-wires prompt instruction, describing the object as puppeteered by thin wires (as shown in Fig. 2 *Automated Demonstration Collection*; additional rollouts are provided in the Appendix), providing a plausible cause that produces clean object-only motion while keeping the object unoccluded.

We further observe three categories of failure in the generated videos: (i) task incompletion, where the grasped object fails to reach the goal configuration; (ii) object duplication or hallucination; and (iii) camera or fixture object drift. To address these, we design a multi-axis VLM [42] rejection filter that scores each candidate video along three axes corresponding to these failure categories. The full prompts remains the same across all tasks, with the task-specific description injected as a placeholder. Videos that do not meet the acceptance threshold are discarded, and the remaining videos are passed to the next step for depth extraction and pose tracking.

b) Depth Extraction and Pose Tracking: We track object poses in the generated videos using FoundationPose [38], which requires per-frame metric depth. We extract depth using Video Depth Anything [43], a video depth estimation model built on the Depth Anything V2 backbone [44], which we prefer over frame-level alternatives for its joint optimization of output quality, temporal consistency, and generalization across diverse scenes.

Video Depth Anything estimates depth in both metric and relative modes. While metric mode is intended to produce depth in physical units, in practice the estimated metric depth deviates significantly from the true metric depth of the conditioning frame. We instead use the relative depth

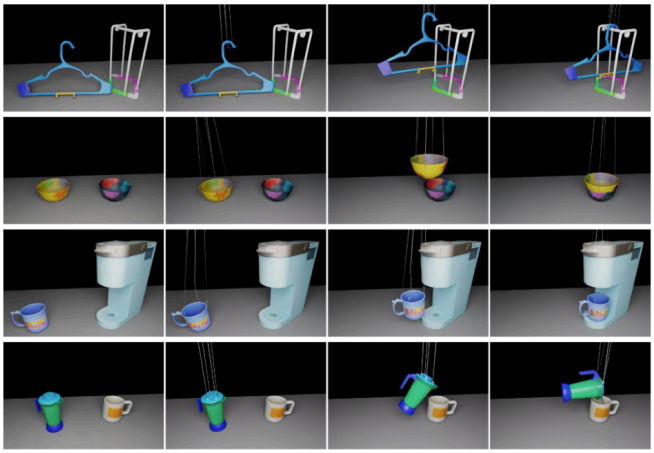


Fig. 4. Examples of generated video sequences for each task. Thin strings descend from above to puppeteer the object, a result of our suspended-wires prompt that yields physically-plausible, occlusion-free object motion.

mode, which produces more robust estimates, and perform a calibration step to ground the relative depth predictions in the physical environment. We recover metric depth from each generated video by solving for a scale s and shift b in disparity space via least-squares regression, fitting $\frac{1}{d_{\text{metric}}} = b + s \cdot d_{\text{relative}}$ using the metric depth of the two objects in the conditioning image $\mathbf{I}_0^{(j)}$, masked to the object regions from Section III-B. We apply the recovered transform to obtain metric-aligned depth across the full video.

To estimate object pose trajectories, we run FoundationPose on each accepted video, providing the metric-aligned depth, per-object segmentation masks, reconstructed meshes $\mathcal{M}_h, \mathcal{M}_f$, and camera intrinsics to obtain per-frame poses $T_{h,t}$ and $T_{f,t}$. FoundationPose occasionally produces axis-flipped initial pose estimates, resulting in incorrect trajectories. To remove such outliers, we compute the Karcher mean $\bar{R}$ of all frame-0 rotation estimates across retained videos and reject any video for which either object's initial rotation deviates from $\bar{R}$ by more than $30°$. We then compute the relative pose trajectory $\tau^{(j)} = (T_t^{\text{rel}})_{t=1}^{T}$ following SPOT [9] and retain the 12 highest-scoring videos per task as training demonstrations.

D. Policy Training and Inference

a) Policy Training: We train SPOT's $\mathrm{SE}(3)$ pose-trajectory diffusion policy [9] on the demonstrations generated in the previous step to obtain policy π. The policy takes the current relative pose T_t^{rel} as input and predicts the future trajectory via $\mathrm{SE}(3)$-equivariant diffusion. We use DDIM as the noise scheduler, with 100 denoising steps during training and 10 steps during inference, and train on a batch size of 128 for 3000 epochs on the 12 synthetically generated demonstrations. The policy predicts over a horizon of 4 future poses; during inference, only the first predicted pose is executed. Following SPOT [9], we additionally train an end-state classifier that predicts task completion from the current relative pose T_t^{rel}.

b) Grasp Proposal and Full-Trajectory Kinematic Feasibility Filter: We automate grasp generation and divide the rollout into three distinct phases: a grasping phase, an object manipulation phase, and a retraction phase.

We generate grasp proposals on o_h using Contact-GraspNet [35]. Since grasping is not learned during policy training, the initial grasp pose is critical to successful execution. To select a feasible grasp, we apply a *full-trajectory kinematic feasibility filter*: we forward-simulate the policy in open loop from the current state to predict the held-object pose sequence $\{T_{h,t}\}$, and compute the corresponding end-effector trajectory for each grasp candidate. Candidates are rejected if the gripper mesh collides with the scene at any predicted step, or if inverse kinematics fails to return a valid joint configuration at any point along the trajectory due to joint limits or kinematic singularities. The surviving candidate with the highest Contact-GraspNet confidence score is selected for execution.

c) Closed-Loop Rollout: Once the grasp is executed, we enter the object manipulation phase, tracking o_h and o_f continuously using FoundationPose [38]. At each timestep, the policy predicts a horizon of four future poses in the fixture object frame; following receding horizon control, only the first is executed by computing the relative $\mathrm{SE}(3)$ transformation to the current pose and mapping it to the end-effector frame via inverse kinematics, regularized to prefer the joint configuration closest to the robot's current one. This closed-loop scheme corrects for accumulated errors and unexpected object movements. The end-state classifier signals task completion when its predicted value exceeds 0.95, upon which the robot releases the object and retracts to its home position.

IV. Experimental Results

Our experiments evaluate three core claims: (1) synthetic demonstrations generated by PROXY are as effective as real human demonstrations, matching SPOT trained on human-hand demonstrations and outperforming Diffusion Policy trained on teleoperation data across four manipulation tasks; (2) the learned policy generalizes across robot embodiments — a Franka Panda and a UR5 — without additional data collection, finetuning, or adaptation; and (3) broadening the scene diversification range recovers performance in challenging settings where real-data baselines degrade, at no additional human cost.

A. Setup

a) Real-World Setup: We evaluate on four tabletop manipulation tasks across two robot embodiments: a Franka Panda with a finray gripper, and a Universal Robots UR5 with a Robotiq 2F-85 parallel-jaw gripper. The four tasks with their corresponding text prompts are: *Pour* (Pour water into the teacup), *Stack* (Stack the yellow bowl inside the blue bowl), *Hook* (Hook the hanger around the top crossbar of the white rack), and *Place* (Place the mug under the coffee machine) as shown in Fig. 6. A fixed ZED2i stereo camera captures the scene for all methods. For PROXY, it provides

the single RGB-D image used for synthetic data generation; for SPOT, it records human demonstration videos. Diffusion Policy additionally uses a wrist-mounted RealSense D435i, with both cameras used for observation during teleoperation across both robot embodiments.

b) Baselines: We evaluate against two baselines, SPOT and Diffusion Policy, using their official implementations. **SPOT** [9] is an object-centric policy that, like PROXY, takes the current relative SE(3) pose between the held and fixture object as input and predicts a trajectory of relative SE(3) poses; we train it on 12 human-hand demonstrations per task, collected under the same scene perturbation range used by PROXY. **Diffusion policy** [1] is an image-based visuomotor policy that maps the RGB observations from the fixed scene and wrist-mounted cameras, along with the current proprioceptive state (end-effector pose), to a sequence of end-effector actions; we train it on 30 teleoperated demonstrations per robot per task. In contrast, PROXY requires only a single RGB-D image to generate all 12 training demonstrations synthetically.

c) Evaluation Protocol: We run 10 trials per method per task on each robot. A trial is counted as a success when the held object reaches the goal configuration relative to the fixture object. We record three categories of failure: *grasp failure*, where the robot fails to acquire the held object; *manipulation failure*, where the robot collides with the scene during execution; and *goal failure*, where the policy fails to reach the desired end configuration.

B. Main Results

We evaluate all methods under two reset distributions. $\mathcal{D}_0$ matches the training distribution, with objects placed within the same spatial range seen during training. $\mathcal{D}_1$ is a more challenging out-of-distribution setting, placing objects over an extended planar region with a wider rotation range. Table I reports per-task success rates at $\mathcal{D}_0$; generalization to $\mathcal{D}_1$ is reported in Section IV-C.

Across all tasks, PROXY matches or exceeds SPOT and substantially outperforms the diffusion policy baseline. The most challenging tasks for both methods were *Hook* and *Pour*, which demand precise pose alignment, as the hanger must be guided onto the rack and the pitcher spout oriented downward, whereas *Stack* and *Place* impose looser alignment constraints. Diffusion policy's comparatively lower success rates reflect its need to learn grasping from scratch alongside task execution, compounded by its sensitivity to initial conditions.

Diffusion policy's performance improves markedly from the Panda to the UR5, as the stiffer UR5 gripper provides more stable object retention than the compliant Panda gripper. The performance gap between PROXY and SPOT comes largely from demonstration quality. SPOT learns from human-hand videos, where the hand occludes the object and degrades pose tracking during training. Since PROXY's synthetically generated videos contain no occlusion, they yield cleaner pose trajectories for training, contributing to its improved performance.

C. Generalization via Training-Distribution Re-Targeting

All policies trained on $\mathcal{D}_0$ degrade when evaluated at $\mathcal{D}_1$ (Table II), with diffusion policy dropping to zero and both SPOT and PROXY dropping to half their success rate. Since our scene diversification is fully synthetic, we introduce **PROXY**$^+$, a variant of **PROXY** trained on 24 synthetic trajectories sampled from a wider scene diversification range ($\pm 0.25\,\mathrm{m}$ in x, $\pm 0.15\,\mathrm{m}$ in y, $\pm 60^\circ$ rotation) at zero additional human cost.

PROXY$^+$ recovers and even surpasses its $\mathcal{D}_0$ performance at $\mathcal{D}_1$, with *Place* improving by over 100% relative to both PROXY and SPOT. This shows that PROXY can be re-targeted to a new deployment distribution simply by regenerating its synthetic training data – an adaptation unavailable to SPOT and Diffusion Policy without collecting additional human or teleoperated demonstrations.

V. Conclusion

We presented PROXY, a system for training object-centric manipulation policies from a single RGB-D image of a scene and a short text description of the task, replacing recorded human demonstrations with synthetic ones generated by our pipeline. On four tasks across two robot platforms, PROXY matches a recent object-centric method trained on real human-hand demonstrations at matched demonstration counts, outperforms Diffusion Policy trained on real teleoperation data, and recovers out-of-distribution performance that real-data baselines lose by re-targeting the synthetic-demonstration distribution at no additional human cost.

VI. Limitations and Future Work

PROXY currently targets two-object tabletop manipulation; extending it to multi-object scenes or non-tabletop settings would require enriching the scene-diversification step. The quality of synthetic demonstrations is bounded by the underlying video generator, so tasks that the generator cannot depict reliably, such as heavy occlusion, deformable objects, or fast dynamics, remain challenging. We see extending to multi-object scenes and cross-instance generalization within an object category as the most natural next steps.

Acknowledgment

We gratefully acknowledge support from NSF grant 2214177; from AFOSR grant FA9550-22-1-0249; from ONR MURI grants N00014-22-1-2740 and N00014-24-1-2603; from the MIT Siegel Family Quest for Intelligence; and from the Robotics and AI Institute.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems (RSS)*, 2023.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems (RSS)*, 2023.

[3] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

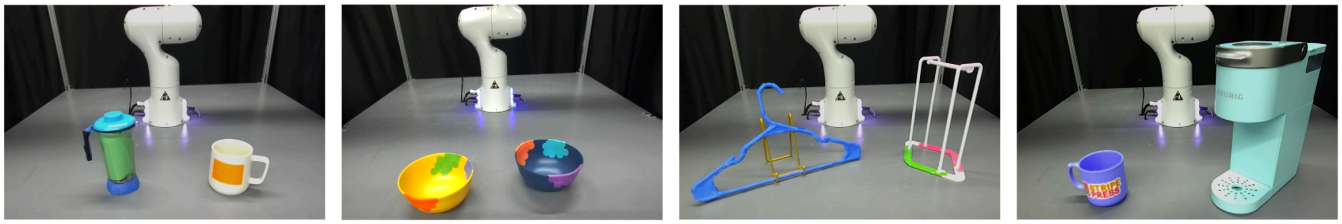


Fig. 5. Overview of the four evaluated tasks (*Pour*, *Stack*, *Hook*, and *Place* and their corresponding data generation text prompts.

TABLE I

MAIN RESULTS AT THE $\mathcal{D}_0$ RESET DISTRIBUTION. SUCCESS RATE OVER 10 TRIALS PER CELL. PROXY AND SPOT EACH USE MATCHED 12 SYNTHETIC AND HUMAN-HAND DEMONSTRATIONS RESPECTIVELY. DP USES 30 TELEOPERATED DEMONSTRATIONS PER ROBOT PER TASK. BEST PER TASK IS HIGHLIGHTED IN BLUE.

	Panda					*UR5*				
Method	Pour	Stack	Hook	Place	*Mean*	Pour	Stack	Hook	Place	*Mean*
DP	0.2	0.4	0	0.1	*0.175*	0.4	0.3	0.5	0.5	*0.425*
SPOT	0.8	0.9	**0.7**	**0.9**	*0.825*	**0.7**	0.7	0.7	0.8	*0.725*
PROXY	**1.0**	**1.0**	0.6	**0.9**	***0.875***	**0.7**	**1.0**	**0.8**	**1.0**	***0.875***

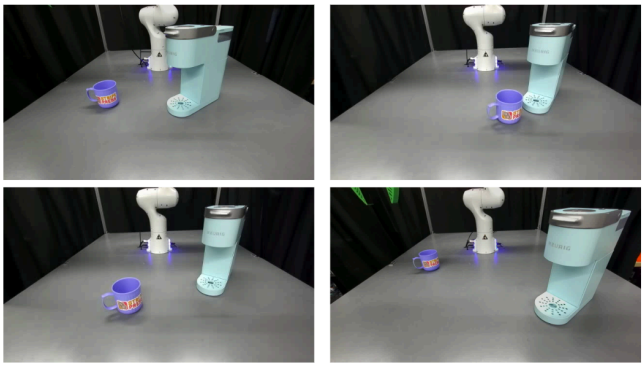


Fig. 6. Comparison of the initial training distribution $\mathcal{D}_0$ (left) and the wider evaluation distribution $\mathcal{D}_1$ (right).

TABLE II

GENERALIZATION TO WIDER RESET DISTRIBUTIONS ON THE PANDA (10 TRIALS PER CELL). ROWS 1–3 REPLICATE TABLE I MODELS; ROW 4 TRAINS ON 24 SYNTHETIC TRAJECTORIES MATCHED TO $\mathcal{D}_1$ AUTOMATICALLY GENERATED, INCURRING NO ADDITIONAL HUMAN COST. BEST PER TASK IS HIGHLIGHTED IN BLUE.

		Pour		*Place*	
Method	# Demonstrations	$\mathcal{D}_0$	$\mathcal{D}_1$	$\mathcal{D}_0$	$\mathcal{D}_1$
DP	30 Teleoperated	0.2	0	0.1	0
SPOT	12 Human	0.8	0.4	0.9	0.3
PROXY	12 Generated	**1.0**	0.5	0.9	0.3
PROXY^{+}	24 Generated	0.9	**0.7**	**0.9**	**0.9**

[4] 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," in *Robotics: Science and Systems (RSS)*, 2024.

[5] H. Fang, H.-S. Fang, Y. Wang, J. Ren, J. Chen, R. Zhang, W. Wang, and C. Lu, "Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 031–15 038.

[6] H.-S. Fang, B. Romero, Y. Xie, A. Hu, B.-R. Huang, J. Alvarez, M. Kim, G. Margolis, K. Anbarasu, M. Tomizuka, E. Adelson, and P. Agrawal, "Dexop: A device for robotic transfer of dexterous human manipulation," *arXiv preprint arXiv:2509.04441*, 2025.

[7] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

[8] 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," 2022.

[9] 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," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[10] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *Advances in neural information processing systems*, vol. 36, pp. 9156–9172, 2023.

[11] 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," *Advances in Neural Information Processing Systems*, vol. 36, pp. 22 304–22 325, 2023.

[12] 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," *arXiv preprint arXiv:2510.08568*, 2025.

[13] B. Chen, T. Zhang, H. Geng, C. Zhang, P. Li, K. Song, W. T. Freeman, J. Malik, P. Abbeel, R. Tedrake, V. Sitzmann, and Y. Du, "Large video planner enables generalizable robot control," *arXiv preprint arXiv:2512.15840*, 2025.

[14] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani, "Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation," in *Conference on Robot Learning (CoRL)*, 2024.

[15] Y. Du, S. Yang, P. Florence, F. Xia, A. Wahid, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, L. Kaelbling, A. Zeng, and J. Tompson, "Video language planning," in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 31 138–31 155.

[16] J. Fu, J. Nan, L. Sun, H. Li, J. Qian, J. L. Barry, K. Ki-

tani, and G. Konidaris, "Novaplan: Zero-shot long-horizon manipulation via closed-loop video language planning," *arXiv preprint arXiv:2602.20119*, 2026.

[17] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators," in *IROS*, 2024.

[18] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song, "Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2025, pp. 437–459.

[19] H. Fang, C. Wang, Y. Wang, J. Chen, S. Xia, J. Lv, Z. He, X. Yi, Y. Guo, X. Zhan *et al.*, "Airexo-2: Scaling up generalizable robotic imitation learning with low-cost exoskeletons," *arXiv preprint arXiv:2503.03081*, 2025.

[20] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "MimicGen: A data generation system for scalable robot learning using human demonstrations," in *Conference on Robot Learning (CoRL)*, 2023.

[21] C. Garrett, A. Mandlekar, B. Wen, and D. Fox, "Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment," *arXiv preprint arXiv:2410.18907*, 2024.

[22] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. J. Fan, and Y. Zhu, "Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 16 923–16 930.

[23] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "RoboCasa: Large-scale simulation of everyday tasks for generalist robots," in *Robotics: Science and Systems (RSS)*, 2024.

[24] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation," *arXiv preprint arXiv:2311.01455*, 2023.

[25] P. Katara, Z. Xian, and K. Fragkiadaki, "Gen2sim: Scaling up robot learning in simulation with generative models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6672–6679.

[26] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to act from actionless videos through dense correspondences," in *ICLR*, 2024.

[27] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "Keto: Learning keypoint representations for tool manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7278–7285.

[28] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," in *Robotics: Science and Systems (RSS)*, 2024.

[29] 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 *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8307–8314.

[30] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," *arXiv preprint arXiv:2302.03573*, 2023.

[31] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023.

[32] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Muller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects," *CVPR*, 2023.

[33] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang *et al.*, "Wan: Open and advanced large-scale video generative models," *arXiv preprint arXiv:2503.20314*, 2025.

[34] S. D. Team, X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, A. Lin, J. Liu, Z. Ma, A. Sagar, B. Song, X. Wang, J. Yang, B. Zhang, P. Dollár, G. Gkioxari, M. Feiszli, and J. Malik, "Sam 3d: 3dfy anything in images," 2025. [Online]. Available: https://arxiv.org/abs/2511.16624

[35] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes," in *ICRA*, 2021.

[36] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[37] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, and others, "Grounded SAM: Assembling open-world models for diverse visual tasks," 2024, arXiv:2401.14159.

[38] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.

[39] E. Coumans and Y. Bai, "PyBullet: A Python module for physics simulation for games, robotics, and machine learning," http://pybullet.org, 2016–2021.

[40] Blender Online Community, "Blender – a 3d modelling and rendering package," http://www.blender.org, Blender Foundation, 2018.

[41] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *ECCV*, 2022.

[42] A. Singh, A. Fry, A. Perelman, A. Tart, A. Ganesh, A. El-Kishky, A. McLaughlin, A. Low, A. Ostrow, A. Ananthram *et al.*, "Openai gpt-5 system card," *arXiv preprint arXiv:2601.03267*, 2025.

[43] 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," *arXiv:2501.12375*, 2025.

[44] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything V2," in *NeurIPS*, 2024.