MATCHINGPOLICY: A CORRESPONDENCE-AWARE POLICY ENABLES CROSS-OBJECT IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context imitation learning enables few-shot policy generalization but struggles to maintain performance on unseen objects and novel scenarios. To address this, we introduce MatchingPolicy, a correspondence-driven framework that explicitly decouples demonstration-to-scene matching from policy learning. Central to our method is a correspondence-aware diffusion policy that conditions robotic actions directly on dense semantic correspondences. This architectural separation reduces the competing demands of correspondence identification and action adaptation, enabling robust out-of-distribution transfer. Our framework integrates vision foundation models with a novel two-stage matching algorithm to dynamically establish reliable correspondences. Extensive evaluations on RLBench and real-world manipulation tasks confirm that MatchingPolicy achieves superior few-shot performance, generalizing reliably across unseen object instances and semantic categories.

1 INTRODUCTION

Imitation learning has demonstrated remarkable potential in acquiring diverse manipulation skills. However, traditional approaches typically require a large amount of costly, task-specific demonstration data to train a policy for even a single skill. While recent efforts have sought to train universal, multi-task imitation policies (Kim et al., 2024) on large and diverse robotic datasets (O'Neill et al., 2024), these models still struggle to generalize to entirely new tasks, and often depend on task-specific fine-tuning to reach acceptable performance. Inspired by breakthroughs in language and vision, in-context learning has recently been introduced to robotic policy learning (Fu et al., 2024) as a way to bypass fine-tuning. In this paradigm, a model is conditioned at test time on a few task demonstrations that provide fine-grained, step-by-step guidance, rather than coarse natural language commands, enabling the policy to adapt to unseen tasks on the fly.

Despite this promise, prior in-context learning approaches for robotics often fail to generalize to novel objects and tasks, even when the demonstrations feature the exact unseen task and object (Jain et al., 2024). We argue that a key reason for this shortcoming is that current methods ask a single model to simultaneously: (1) infer precise correspondences between the demonstration and the current scene from scratch using limited robotic-domain data, and (2) adapt the demonstrated motions to the new objects and scenes. This dual burden can hinder both correspondence extraction and action adaptation, especially in visually diverse or geometrically complex environments.

To address this challenge, we propose MatchingPolicy, a novel in-context imitation learning framework for robotic manipulation that explicitly decouples correspondence extraction from policy learning. Rather than forcing the policy to implicitly discover correspondences, we leverage explicitly precomputed correspondences and focus on training a policy that adapts robot behavior accordingly. Additionally, we introduce an online adaptive semantic matching algorithm that dynamically establishes dense and reliable correspondences between demonstrations and the current scene.

Our correspondence-aware policy offers three key advantages. First, by leveraging explicit correspondences between the demonstration and the current scene, the policy can more precisely localize task-relevant interaction regions, enabling more accurate and adaptive action generation. Second, as the policy operates on correspondence features rather than raw visual object representations, it reduces dependence on object-specific appearance, enhancing its ability to generalize across different

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

object instances and categories. Third, the decoupled design, separating correspondence extraction from policy learning, facilitates the seamless integration of powerful off-the-shelf visual and semantic matching methods (Oquab et al., 2023; Zhang et al., 2023) without retraining.

Our results show that the proposed MatchingPolicy outperforms the state-of-the-art in-context imitation learning method on RLBench (James et al., 2020). We further validate our approach on a set of challenging real-world manipulation tasks, where it demonstrates strong generalization to a wide variety of objects. The main contributions are as follows:

- We propose a novel approach that separates cross-scene correspondence extraction from policy learning, empowered by a correspondence-aware policy conditioned on the extracted explicit correspondences, thereby enhancing flexibility and scalability.
- We design a novel diffusion policy biased to leverage direct relationships from explicit correspondences, improving the model's generalization across tasks and objects.
- We introduce an online adaptive matching algorithm that dynamically provides reliable correspondences during execution, enabling precise real-world policy adaptation.
- We conduct extensive evaluations of our method in both simulation and real-world settings. The results demonstrate that the proposed approach achieves strong few-shot performance, while also exhibiting robust cross-instance and even cross-category generalization.

2 Related Work

2.1 Multi-task Imitation Learning

A growing body of work focuses on developing generalist robot policies (Kim et al., 2024; Team et al., 2024; Jang et al., 2022; Black et al., 2024; Liu et al., 2024; Cheang et al., 2024; Lin et al., 2026). These models, typically conditioned on visual observations and language instructions, are trained on large-scale robotic datasets in the real world (O'Neill et al., 2024; Khazatsky et al., 2024; Bu et al., 2025) or in simulation (James et al., 2020; Mu et al., 2024) and achieve strong performance on the tasks seen during training. With appropriate visual inputs and language prompts, they can generalize zero-shot to related tasks and scenes under varying visual conditions. However, their performance often degrades on entirely unseen tasks; additional fine-tuning is usually required to attain acceptable results. Our method bypasses the task generalization obstacle by directly conditioning the policy on demonstrations, which showcase the fine-grained intention of the task more effectively.

2.2 In-context Imitation Learning

In-context learning (ICL) has emerged as a powerful paradigm, enabling models to adapt to novel tasks in language (Brown et al., 2020), vision (Zhou et al., 2024) and sequential decision making (Raparthy et al., 2023) from only a handful of demonstrations without retraining. In robotics, explicit ICL-based approaches often begin by establishing correspondences between given demonstrations and a new scene, followed by transforming the trajectories using handcrafted heuristics (Zhang & Boularias, 2024; Tang et al., 2025; Zhu et al., 2024; Heppert et al., 2024). While effective in controlled settings, they typically demand considerable human effort and their adaptability to truly novel scenarios remains constrained. The in-context capability of LLMs has also been adapted to robotic action prediction (Di Palo & Johns, 2024; Yin et al., 2024). However, these methods usually operate in an open loop and struggle to predict accurate motions efficiently.

Recent studies have pursued training in-context policy models directly, leveraging transformers (Jain et al., 2024; Fu et al., 2024) or graph neural networks (Vosylius & Johns, 2024) to minimize human involvement. In contrast, MatchingPolicy adopts a direct and modular strategy: it explicitly extracts correspondence information from vision foundation models (VFMs), freeing the policy to focus solely on adapting robot actions to the new environment. This decoupled design not only reduces manual intervention but also enhances generalization to novel objects.

Recent works by Sridhar et al. (2024; 2025) employ a similar modular design, decoupling relationship extraction between demonstrations and the current scene from action prediction using retrieval-augmented generation. In contrast, our method extracts dense pointwise correspondences, which provide both a similarity measure and fine-grained spatial cues for robot action prediction.

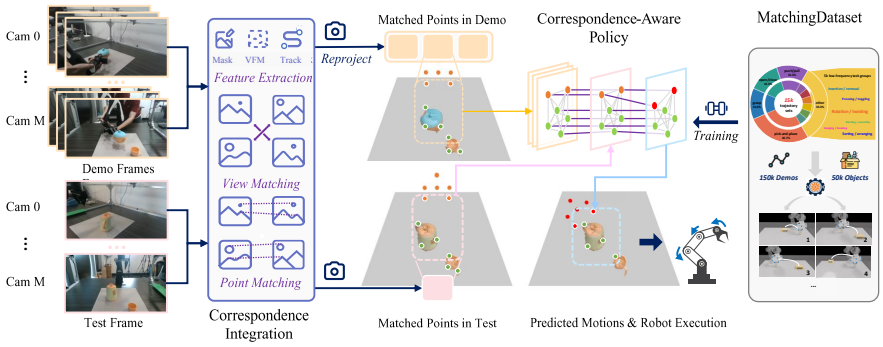

Figure 1: MatchingPolicy. Two-stage VFM matching extracts correspondences for a graph-based diffusion policy predicting 6D gripper poses. The policy is trained on synthetic data with ground-truth correspondences.

2.3 Keypoint-based Policy Learning

Keypoint-based representations are widely used in robotics because of their sparsity and interpretability (Puang et al., 2020; Gao et al., 2023). They also serve as high-level abstract action representations, either predicted by large language models (LLMs) (Huang et al., 2024a;b) or generated by learned policies (Wen et al., 2023; Bharadhwaj et al., 2024; Xu et al., 2024), and can be mapped to low-level robotic control, enabling LLM-driven manipulation and cross-embodiment policy transfer.

Recently, several works have integrated keypoint-based representations into policy training as observations (Haldar & Pinto, 2025; Wang et al., 2024; Levy et al., 2024; Wang et al., 2025; Fang et al., 2025), demonstrating improved generalization to novel scenes and objects and improving data efficiency. These methods typically rely on predefined keypoints, either selected manually or determined via unsupervised clustering, which can limit the policy's adaptability to novel scenarios. Our work differs in two key aspects: 1) Task Scope: our focus is adaptation to new tasks through demonstration context at test time, without task-specific parameter updates; 2) Keypoint Adaptation: instead of static predefined keypoints, our method dynamically generates keypoints based on matching confidence and visibility. This reduces manual keypoint annotation while improving feature alignment.

3 Method

Problem Formulation. We address robotic manipulation in a few-shot, in-context learning setting. Given a small set of N demonstrations $\{D_j\}_{j=1}^{N}$, where $N \in \{1, 2\}$, the goal is to leverage these demonstrations to perform the same task depicted in the demos, but in a novel scene. The policy observation of the current scene is represented by a state $s_c = (P_c, g_c)$. Here, P_c is the RGB-D point cloud expressed in the current gripper frame using T_c^{-1}, where $T_c \in \mathrm{SE}(3)$ maps gripper coordinates to world coordinates; $g_c \in \{0, 1\}$ is the gripper state. Similarly, a demonstration D_j is a time-indexed sequence of L states, $D_j = (s_{j,t})_{t=1}^{L}$, where each $s_{j,t} = (P_{j,t}, g_{j,t})$ follows the same definition as s_c. Given the current state s_c and the demonstration set $\{D_j\}_{j=1}^{N}$, the policy π predicts a horizon of K future actions $(a_k)_{k=1}^{K}$, where $a_k = (\Delta T_k, g_k)$ comprises a gripper transformation $\Delta T_k = T_c^{-1} T_k$ relative to the current gripper frame and a target gripper state g_k; T_k is the future gripper pose in world coordinates.Formally, the objective is to learn a policy that models the conditional distribution $\pi = p(a_{1:K} \mid s_c, (D_j)_{j=1}^{N})$. For ease, we use the term "frame" to refer uniformly to the current state, a demonstration state, or a predicted action, drawing a direct analogy to frames in a video.

Method Overview. The pipeline of our method is shown in Figure 1. Our key insight is to decouple correspondence extraction from action prediction. Rather than requiring the policy to infer demonstration-current alignment implicitly, we first explicitly establish semantic correspondences between the current observation and the demonstration frames. Based on these correspondences, we select a sparse set of keypoints in the current frame together with their matched counterparts across the demonstrations. These correspondence-indexed point sets provide a structured context that captures task-relevant spatial and temporal cues from the demonstrations. Given this structured context, our correspondence-aware policy predicts future end-effector motions for the current scene. Correspondences serve as an explicit prior for routing information from demonstrations to the current

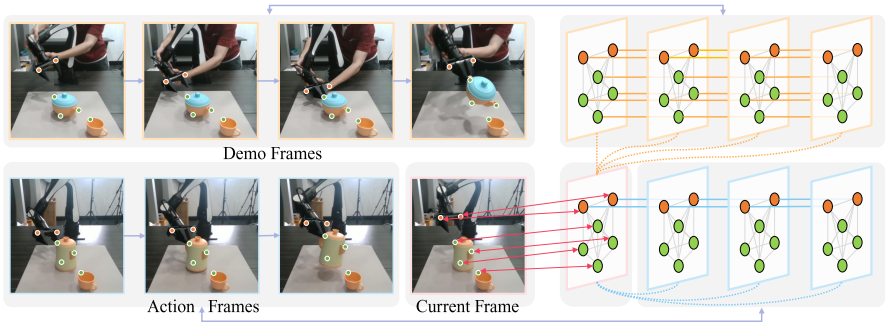

Figure 2: Correspondence-aware policy. Left: matched keypoints in demonstration, current, and action frames. Right: the policy graph. Red two-way arrows map current-frame keypoints to graph nodes; graph edges carry information. Inter-frame links are simplified for clarity.

observation and future action predictions, enabling actions consistent with the demonstrated intent. The predicted end-effector motions are converted into executable actions with inverse kinematics.

We use our synthetic demonstration dataset with ground-truth correspondences (Section 3.2) to train the correspondence-aware policy (Section 3.1). At deployment, where ground-truth correspondences are unavailable, we combine vision foundation models with visual point trackers to extract semantic and spatio-temporal correspondences from real-world observations (Section 3.3). The same policy uses these correspondences without changing its action prediction module.

3.1 Correspondence-Aware Diffusion Policy

Keypoint Selection with Correspondence. To reduce network burden and improve robustness, we select scene keypoints from the current frame by considering both geometric coverage and correspondence reliability across demonstrations. Specifically, a point is retained as a candidate only if it has semantically matched counterparts in most demonstration frames, which filters out unstable matches caused by occlusions or errors from vision foundation models. We then apply Farthest Point Sampling (FPS) to the candidate set to obtain a spatially representative and diverse subset of scene keypoints, and select their matched points in the demonstration frames accordingly.

In addition to scene keypoints, we include a fixed set of gripper keypoints to represent the gripper pose and state. Unlike scene keypoints, gripper keypoints have known one-to-one correspondences across frames by construction. Together, the selected scene and gripper keypoints form correspondence-indexed entities: each selected current-frame keypoint is associated with matched demonstration keypoints under the same correspondence identity. These identities serve as the structural prior for our correspondence-conditioned policy.

Correspondence-Conditioned Action Prediction. Our policy uses the correspondence-indexed entities as an explicit structural prior for action prediction. Although the estimated correspondences may still be imperfect, they provide strong cues about which parts of the demonstrations should influence the current scene. We organize the input around these correspondence identities. Specifically, we group nodes into demonstration frames $\{D_j\}_{j=1}^{N}$, the current frame s_c, and K action frames $(a_k)_{k=1}^{K}$, where action frames represent future gripper poses and states. Each frame contains $M_g = 6$ gripper nodes and $M_s = 16$ scene nodes. Nodes sharing the same index are treated as corresponding entities: the m-th scene node in a demonstration frame and the current frame refer to matched scene keypoints, while the m-th gripper node across current and action frames refers to the same gripper keypoint. In demonstration and current frames, gripper nodes store 3D gripper-keypoint positions and the gripper state, while scene nodes store selected scene-keypoint positions. In action frames, gripper nodes store predicted keypoint positions and gripper states relative to the current frame, and scene nodes are copied from the current frame as a fixed scene reference.

Since the representation consists of sparse keypoints with structured spatial, temporal, and correspondence relations, a graph with message passing naturally routes information among them, as shown on the right of Figure 2. The information flow is defined by three types of relations. First, cross-frame correspondence edges connect each current-frame node to its corresponding nodes in

the demonstration frames, and also connect corresponding nodes between adjacent demonstration frames. These orange edges in Figure 2 provide direct pathways for transferring demonstrated intent through matched keypoints. Second, intra-frame spatial edges fully connect gripper and scene nodes within each demonstration, current, and action frame. These gray edges model each frame's spatial configuration and allow correspondence information received by scene nodes to propagate to gripper nodes, which are then used for action prediction. For these spatial edges, we use the relative Cartesian displacement between nodes as the edge attribute. Third, action-frame consistency edges connect gripper nodes in each action frame to their corresponding nodes in the current frame, as well as preceding action frames. These blue edges anchor future gripper keypoints to the current observation and encourage temporally consistent predictions. After predicting gripper-keypoint displacements, we recover the relative gripper pose transformation ΔT_k for each action frame using SVD (Arun et al., 1987). Overall, correspondences are routed from demonstrations to the current scene, grounded through scene-gripper spatial relations, and decoded into future gripper motion.

Local Geometric Refinement. The selected keypoints provide only a sparse scene representation, and the estimated correspondences can be noisy or ambiguous. We attach local geometric evidence to each selected scene keypoint. Each unselected point in the point cloud is assigned to its nearest selected keypoint, forming local point groups around correspondence-indexed scene nodes. These groups are processed by a local PointNet++ encoder (Qi et al., 2017), pre-trained with a geometric reconstruction objective (Mescheder et al., 2019). The resulting features summarize fine-grained local shape evidence around each scene keypoint. Local shape features are propagated to gripper nodes through spatial edges, providing geometric context alongside the correspondence-indexed keypoints.

Diffusion-based Action Decoding. We predict translation- and rotation-induced displacements separately. For canonical gripper keypoint q_i, let $(R_{k,0}, t_{k,0})$ and $(R_{k,\tau}, t_{k,\tau})$ be the clean and noisy relative poses of action k at diffusion step τ. The geometric targets are

$$d^{\text{trans}}_{k,i,\tau} = t_{k,0} - t_{k,\tau}, \qquad d^{\text{rot}}_{k,i,\tau} = (R_{k,0} - R_{k,\tau})q_i.$$

Together with a scalar gripper-state residual, they form a 7D target per gripper node, supervised by MSE (Ho et al., 2020). The two geometric components sum to the clean-minus-noisy point displacement; denoising combines them and recovers rigid pose updates by SVD (Arun et al., 1987). Appendix C details the denoising and rigid alignment, and training details can be found in E.

3.2 MATCHINGDATASET: CORRESPONDENCE-RICH IN-CONTEXT LEARNING DATA

To train an in-context imitation policy, we require intent-consistent trajectories across diverse scene configurations, together with 3D correspondences across the scenes and trajectories. Existing robot datasets, however, are limited in task and object diversity. We therefore design MATCHINGDATASET based on two observations: (i) the policy mainly relies on the relationship between the demonstration and the target scene, rather than object geometry or realistic task appearance; and (ii) cross-scene correspondences are readily available in simulation due to accessible ground-truth poses. As a result, we build a large-scale synthetic manipulation dataset with 15K intent-consistent task sets, over 150K trajectories, and more than 50K manipulated-object instances. The dataset uses task-relevant meshes sampled from ShapeNet (Chang et al., 2015) and covers both common manipulation behaviors and diverse low-frequency tasks under varied scene configurations, as shown in Figure 3.

Each trajectory set is initialized from a procedurally constructed reference demonstration. The variant object-centric trajectories are spatially retargeted by adjusting object poses and scene layouts, while preserving the robot-object relative poses at key frames. As a result, the generated trajectories exhibit diverse observations, robot motions, and spatial configurations, yet maintain the same underlying task intent. Observations are captured as RGB-D point clouds using virtual cameras to mimic realistic manipulation scenarios. For each object mesh, we pre-sample a dense surface point template in its local coordinate frame, which serves as a canonical representation shared by all trajectories in the same set. Since ground-truth object poses are available in simulation, the template can be transformed into each scene and time step, allowing observed object points to be associated with their corresponding template points. Cross-scene and cross-trajectory 3D correspondences are then computed through this shared template space. Some template points may not find valid matches in certain time steps in demonstrations within the distance threshold, mimicking real-world partial observations. Each trajectory contains synchronized RGB-D observations, point clouds, and the associated correspondence annotations. Data generation and annotation details are in Appendix A.

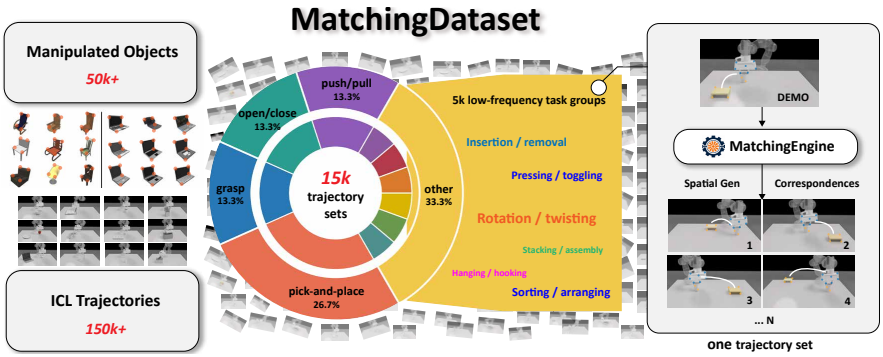

Figure 3: Overview of MATCHINGDATASET. MATCHINGENGINE spatially retargets each reference demonstration to produce task-consistent trajectories with ground-truth 3D correspondences.

3.3 REAL-WORLD CORRESPONDENCE INTEGRATION VIA VFMS

An advantage of our method is its plug-and-play ability to incorporate off-the-shelf vision foundation models at test time. Given demonstrations and a novel scene of the same task, we sample object-centric candidate points in the initial observations, match them across scenes using VFM features (Heinrich et al., 2025), propagate the matched points over time with a point tracker (Karaev et al., 2024), and lift valid 2D matches and tracks to a shared 3D world frame for action prediction.

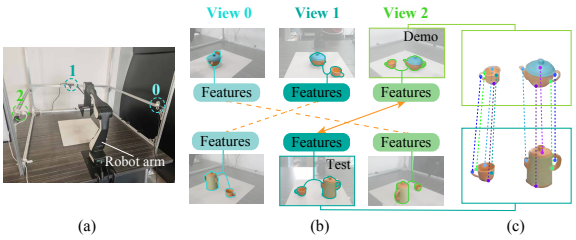

Figure 4: Real-world matching: (a) camera setup; (b) view alignment using pooled object features; (c) point matches between aligned views.

Inter-scene Correspondence via Semantic Matching. Our two-stage matcher first aligns camera views, then matches object points within aligned views, following recent VFM-based methods (Zhang et al., 2023). Objects masks are obtained using Segment Anything (Kirillov et al., 2023). For each RGB-D observation, we extract dense descriptors within these masks and L_2-normalize them before matching. Given a pair of aligned views, we compute 2D correspondences by nearest-neighbor search in descriptor space. For a candidate pixel u in one masked image, its correspondence in the paired image is $v^* = \arg\max_{v \in M_b} F_a(u)^\top F_b(v)$, where F_a and F_b are the normalized dense descriptors of the two images, and M_b is the target object mask in the paired image. The resulting matches provide semantic 2D correspondences between demonstration and current scenes. For multi-object tasks, we apply the same procedure independently within each object mask. Single-view matching can be unreliable under large viewpoint or object-pose changes due to occlusions and appearance variations. To improve robustness, we perform adaptive view alignment before dense matching. Our setup uses three calibrated RGB-D cameras covering approximately 360°, providing complementary observations of the scene. Rather than assuming fixed camera-index correspondences, we compare pooled view-level descriptors and select the most compatible view assignment. For a candidate assignment σ, we compute

$$S(\sigma) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \bar{F}_{a,i}^\top \bar{F}_{b,\sigma(i)},$$

where $\mathcal{V}$ indexes camera views and $\bar{F}_{a,i}$ and $\bar{F}_{b,\sigma(i)}$ are pooled descriptors from the paired views. We choose $\sigma^* = \arg\max_\sigma S(\sigma)$ by enumerating feasible assignments, and perform dense matching only on the top two aligned view pairs under σ^*. Although our experiments use three cameras, the formulation naturally extends to general multi-view settings.

Intra-scene Correspondence via Visual Tracking. Following prior methods (Haldar & Pinto, 2025; Wang et al., 2025), we use an off-the-shelf point tracker (Karaev et al., 2024) to propagate correspondences over time. Starting from the matched object points in the initial frames, we track their 2D locations through each demonstration sequence offline and through the online execution

Table 1: Success rates on 36 RLBench tasks, reported as NoFT/FT (without/with fine-tuning on Adaptation Tasks). OOD-Easy/Hard have similar/dissimilar interaction patterns. IP: InstantPolicy; MP: MatchingPolicy.

Adaptation Tasks			**OOD-Easy Tasks**			**OOD-Hard Tasks**		
Task	IP	MP	Task	IP	MP	Task	IP[1]	MP
Open Box	0.94/0.99	1.00/1.00	Slide Buzzer	0.35/0.94	0.71/0.74	Close Drawer	–/0.00	0.87/0.90
Close Jar	0.58/0.93	0.78/0.92	Plate Out	0.81/0.97	0.96/0.99	Close Grill	–/0.00	0.98/0.97
Toilet Seat Down	0.85/0.93	1.00/1.00	Close Laptop	0.91/0.95	1.00/1.00	Open Grill	–/0.00	0.97/0.99
Close Microwave	1.00/1.00	0.97/1.00	Close Box	0.77/0.99	1.00/1.00	Pick Up Cup	–/0.00	0.90/0.94
Phone on Base	0.98/1.00	1.00/1.00	Open Jar	0.52/0.78	0.89/0.91	Open Door	–/0.01	0.99/0.98
Lift Lid	1.00/1.00	1.00/1.00	Toilet Seat Up	0.94/1.00	1.00/1.00	Turn Tap	–/0.00	0.98/1.00
Umbrella Out	0.88/0.91	0.92/1.00	Meat off Grill	0.77/0.90	0.82/0.89	Frame Off	–/0.00	0.35/0.33
Slide Block	0.75/1.00	0.98/1.00	Open Microwave	0.23/0.56	0.43/0.54	Sweep Dust	–/0.00	0.37/0.41
Push Button	0.60/1.00	0.89/1.00	Paper Roll Off	0.70/0.95	0.94/0.96	Open Wine	–/0.01	0.69/0.71
Basketball	0.66/0.97	0.87/0.97	Put Rubbish	0.97/0.99	1.00/1.00	Put Money	–/0.00	0.52/0.50
Meat on Grill	0.78/1.00	0.83/1.00	Put Umbrella	0.31/0.37	0.33/0.40	Put Roll	–/0.00	0.49/0.52
Flip Switch	0.40/0.94	0.90/0.99	Lamp On	0.42/0.41	0.81/0.85	Pull Plug	–/0.00	0.48/0.46
Average	0.79/0.97	0.92/**0.99**	**Average**	0.64/0.82	0.83/**0.86**	**Average**	–/0.00	0.72/**0.73**

[1] IP results for Adaptation/OOD-Easy are from Vosylius & Johns (2024); OOD-Hard uses the official RLBench-fine-tuned checkpoint, evaluated locally. NoFT results are unavailable because the corresponding weights were not available.

sequence at test time. Unlike approaches that rely on sparse, expert-annotated, or clustered keypoints, we track a set of object-centric points, with $N_{\text{track}} = 512$ points per object and camera view. Tracking runs independently per camera view. For each point, the tracker predicts its 2D location and visibility per frame, producing temporally consistent image-space trajectories for 3D lifting.

Geometry-based 3D Lifting and Point Filtering. Valid matches and tracks are back-projected to 3D using depth and camera calibration, then transformed into the corresponding gripper frame. Appendix D details depth and visibility filtering.

4 Experiments

In this section, we address the following questions: 1) Does MatchingPolicy outperform baseline methods overall as it can better utilize explicit correspondences? 2) Can MatchingPolicy leverage an off-the-shelf large vision model to facilitate generalization in real-world scenarios?

4.1 Simulation Experiments

Evaluation Settings. We first evaluate MatchingPolicy on 36 RLBench (James et al., 2020) simulation tasks covering diverse objects, initial configurations, and interaction patterns. We divide these tasks into three groups: 1) 12 **Adaptation Tasks**, which are used for fine-tuning to reduce the domain gap between synthetic pretraining and RLBench; 2) 12 **OOD-Easy Tasks**, which are excluded from fine-tuning but share similar interaction structures with the Adaptation Tasks (e.g., closing a box versus opening a box); and 3) 12 **OOD-Hard Tasks**, which are also excluded from fine-tuning but involve less familiar interaction patterns, object affordances, or task semantics. We compare MatchingPolicy with three representative approaches: InstantPolicy (Vosylius & Johns, 2024), a geometric in-context learning method based on graph neural networks; KAT (Di Palo & Johns, 2024), a keypoint-based in-context learning method that uses LLM reasoning for action generation; and RDT-1B (Liu et al., 2024), a representative Vision-Language-Action (VLA) model. Ground-truth correspondences are obtained following the procedure described in Section A. For each method and setting evaluated locally, the evaluation run includes 100 episodes per task using a fixed random seed.

Results. We first compare MatchingPolicy with InstantPolicy on all 36 RLBench tasks, using two in-context demonstrations for both methods. As summarized in Table 1, MatchingPolicy achieves higher average success rates across all three task groups. A key strength of MatchingPolicy is its ability to generalize without task-specific fine-tuning. On several unseen tasks, such as *Open Jar* and *Lamp On*, it even outperforms the baseline after the latter is fine-tuned on the Adaptation Tasks. This advantage is most pronounced on OOD-Hard, where the fine-tuned baseline achieves near-zero performance even when some of the relevant objects have already appeared during adaptation. We hypothesize that the baseline's strong reliance on object geometry makes it difficult to infer task intent

when interaction patterns change. In contrast, MatchingPolicy conditions directly on demonstration-to-scene correspondences, enabling it to transfer interaction patterns with less dependence on exact object geometry and generalize more effectively to unseen tasks.

We compare KAT and RDT-1B on six selected tasks (Table 2), given the cost of task-specific baseline fine-tuning. All results are evaluated in our environment. KAT uses the same two demonstrations as Table 1, or ten following its original demonstration budget (Di Palo & Johns, 2024). RDT-1B uses its pretrained checkpoint, or is fine-tuned on 20 demonstrations per task following its original budget (Liu et al., 2024). MatchingPolicy uses two in-context demonstrations without RLBench fine-tuning. Using only two demonstrations, MatchingPolicy consistently outperforms KAT, even when KAT is provided with ten demonstrations. It also achieves higher success rates than RDT-1B, even after the latter is fine-tuned on 20 demonstrations. Together, these results highlight the effectiveness and demonstration efficiency of explicit correspondence modeling compared with LLM-based in-context learning and VLA fine-tuning.

Table 2: Six-task RLBench success rates.

KAT and MP use GT correspondences. RDT-1B 20-Shot uses task-specific fine-tuning.

	KAT		RDT-1B		MP
Task	2-Shot	10-Shot	0-Shot	20-Shot	2-Shot
Plate Out	0.12	0.36	0.08	0.67	**0.96**
Slide Buzzer	0.04	0.19	0.00	0.47	**0.71**
Toilet Seat Up	0.25	0.38	0.12	0.69	**1.00**
Meat off Grill	0.33	0.54	0.05	0.73	**0.82**
Pick Up Cup	0.28	0.47	0.09	0.81	**0.90**
Pull Plug	0.00	0.12	0.00	0.32	**0.48**
Average	0.17	0.34	0.06	0.62	**0.81**

4.2 Real-World Experiments

Setup and Evaluation Settings. We evaluate MatchingPolicy on four real-world manipulation tasks spanning rigid-object and articulated-object manipulation. We use an AirBot Pro 6-DoF robotic arm, with the scene captured by three Intel RealSense D435i RGB-D cameras. We deploy the MatchingDataset-pretrained policy without RLBench or real-world fine-tuning, using two in-context demonstrations per task and visually estimated correspondences (Section 3.3). To rigorously evaluate generalization, we consider three levels of difficulty: **Layout-Easy**, which uses the same object instances as the demonstrations with minor pose variations; **Layout-Hard**, which uses the same instances with substantial pose variations, including markedly different orientations; and **Shape**, which uses novel object instances that differ in appearance and geometry. For each task and evaluation setting, we conduct ten trials and report the average success rate. See Section B for task illustrations.

As shown in Table 3, MatchingPolicy substantially outperforms InstantPolicy across all three evaluation settings. Its performance remains robust under both layout and object variations, demonstrating strong real-world generalization without real-world fine-tuning.

Generalization to Novel Layouts and Objects. We observe that the success rate of InstantPolicy on the *Put Lid* task in the **Layout-Hard** setting drops to 0%. Relying on a coarse geometric representation and biased by the demonstration trajectory, it fails by grasping the pot instead of the lid. In contrast, our approach leverages VFMs to extract precise semantic correspondences, enabling MatchingPolicy to identify the lid and complete the task. In the more challenging task *Open Box*, MatchingPolicy can generate successful trajectories approaching objects from directions very

Table 3: Real-world generalization of InstantPolicy (IP) and MatchingPolicy (MP): successes out of ten trials and average success rates (%).

	Layout-Easy		Layout-Hard		Shape	
Task	IP	MP	IP	MP	IP	MP
Pour Water	0/10	**7/10**	0/10	**6/10**	0/10	**8/10**
Put Lid	**9/10**	8/10	0/10	**6/10**	4/10	**6/10**
Open Box	0/10	**9/10**	0/10	**8/10**	0/10	**8/10**
Cut Egg	0/10	**7/10**	0/10	**6/10**	0/10	**7/10**
Avg. (%)	22.5	**77.5**	0.0	**65.0**	10.0	**72.5**

Table 4: Matching and planning ablations in the real world. Entries report success rates (%) under the same evaluation settings as Table 3.

Matching	Naive	Two-stage		
Planning	Overlap	Whole	Half	Overlap
Layout-Easy	60.0	55.0	60.0	**77.5**
Layout-Hard	20.0	47.5	45.0	**65.0**
Shape	45.0	57.5	45.0	**72.5**

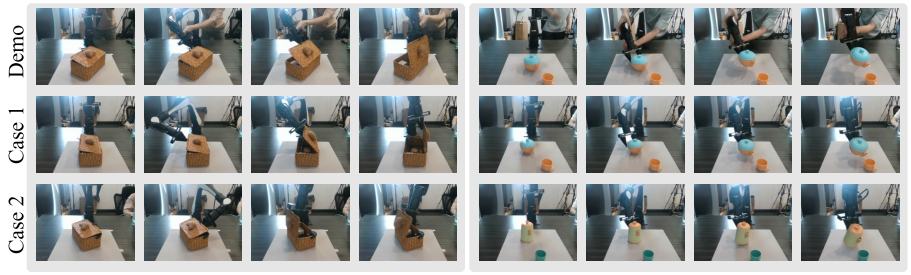

Figure 5: Generalization from two in-context demonstrations
. Left: novel layouts; right: unseen object instances.

different from those in the demonstration (Figure 5). This shows our method follows the motion of matched keypoints instead of merely replaying the demonstrated trajectory.

In the **Shape** generalization setting, MatchingPolicy substantially outperforms the baseline, achieving a 72.5% average success rate. As shown on the right of Figure 5, MatchingPolicy completes the *Pour Water* task even when the teapot differs substantially in shape. The policy demonstrates cross-category generalization on the *Put Lid* task, correctly placing a lid on a teapot using demonstrations of placing a lid onto a pot (see Figure 9). These results demonstrate generalization across object geometries.

4.3 Ablation Studies

Two-Stage Matching versus Naive 2D-to-3D Matching. We compare our two-stage matching approach against a 3D naive matching baseline. The baseline, based on Wang et al. (2024), projects 2D features into the world frame and fuses them by their 3D coordinates before establishing correspondences. As shown in Table 4, our method achieves superior performance by generating more accurate semantic correspondences. We observe that the performance of the naive baseline drops in the **Shape** and **Layout-Hard** settings, as naive matching produces many incorrect correspondences which mislead the policy and lead to task failure.

Test-time Planning Methods. We develop a planning strategy adapted from Janner et al. (2022) to improve test-time performance: After the policy predicts K actions, the robot executes the first $K/2$ actions. The remaining $K/2$ actions seed the next prediction by replacing the first $K/2$ actions of the next action chunk, resulting in an inference process where the first half of an action chunk is denoised for only the final step, while the second half undergoes the full diffusion process. We term this strategy "overlapping chunks" and use it in other real-world experiments. We compare it against several alternatives: 1) Whole chunk ("Whole"): the robot executes all K predicted actions. 2) Half chunk ("Half"): the robot executes the first $K/2$ steps and then re-predicts. Table 4 shows that "overlap" substantially enhances policy performance. We also observe smoother trajectories.

5 Conclusions and Limitations

Conclusions. In this work, we introduced MatchingPolicy, a novel correspondence-aware policy that transforms robot actions from demonstrations to new scenes using dense correspondence cues. By leveraging off-the-shelf vision foundation models with a novel two-stage matching algorithm, our approach enables few-shot in-context imitation on tasks with unseen objects and different layouts in the real world. Our work demonstrates that explicitly modeling the visual and semantic relationships between the demonstration and the current scene is key to effective in-context imitation learning. This approach enables adaptation to new scenes from a small demonstration context.

Limitations. Our method also has limitations. First, we focus primarily on short-horizon manipulation tasks; extending our approach to long-horizon tasks would be a valuable direction for future work. Second, our method considers semi-dynamic scenes and operates at a relatively slow speed, making it difficult to handle highly dynamic tasks such as cloth flinging (Ha & Song, 2022). Third, our view selection process depends on our current platform, requiring adaptation on other platforms.

AI USE STATEMENT

We used generative AI tools to assist with implementing some utility functions in the project code. The overall framework was implemented by the authors. During manuscript preparation, we also used these tools for feedback on methodological descriptions, interpretation and presentation of experimental results, language editing, LaTeX formatting, figure-label revisions, and bibliographic and consistency checks. The authors take responsibility for the accuracy and integrity of the final paper and associated code, including all AI-assisted content.

REFERENCES

K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Norman Di Palo and Edward Johns. Keypoint action tokens enable in-context imitation learning in robotics. *arXiv preprint arXiv:2403.19578*, 2024.

Xiaolin Fang, Bo-Ruei Huang, Jiayuan Mao, Jasmine Shone, Joshua B Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. KALM: Keypoint abstraction using large models for object-relative imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8307–8314. IEEE, 2025.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024.

Jianfeng Gao, Zhi Tao, Noémie Jaquier, and Tamim Asfour. K-VIL: Keypoints-based visual imitation learning. *IEEE Transactions on Robotics*, 39(5):3888–3908, 2023.

Huy Ha and Shuran Song. FlingBot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pp. 24–33. PMLR, 2022.

Siddhant Haldar and Lerrel Pinto. Point Policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved baselines for agglomerative vision foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22487–22497, 2025.

Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. DITTO: Demonstration imitation by trajectory transformation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7565–7572. IEEE, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. CoPa: General robotic manipulation through spatial constraints of parts with foundation models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9488–9495. IEEE, 2024a.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024b.

Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, et al. Vid2Robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Mara Levy, Siddhant Haldar, Lerrel Pinto, and Abhinav Shirivastava. P3-PO: Prescriptive point priors for visuo-spatial generalization of robot policies. *arXiv preprint arXiv:2412.06784*, 2024.

Haitao Lin, Hanyang Yu, Jingshun Huang, He Zhang, Yonggen Ling, Ping Tan, Xiangyang Xue, and Yanwei Fu. PoseVLA: Universal pose pretraining for generalizable vision-language-action policies. In *Robotics: Science and Systems*, 2026. URL https://arxiv.org/abs/2602.19710.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864, 2023.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision*, pp. 264–273. Springer, 2024.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

En Yen Puang, Keng Peng Tee, and Wei Jing. KOVIS: Keypoint-based visual servoing with zero-shot sim-to-real transfer for robotics manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7527–7533. IEEE, 2020.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017.

Sharath Chandra Raparthy, Eric Hambro, Robert Kirk, Mikael Henaff, and Roberta Raileanu. Generalization to new sequential decision making tasks with in-context learning. *arXiv preprint arXiv:2312.03801*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. URL https://arxiv.org/abs/2010.02502.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. REGENT: A retrieval-augmented generalist agent that can act in-context in new environments. *arXiv preprint arXiv:2412.04759*, 2024.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. RICL: Adding in-context adaptability to pre-trained vision-language-action models. *arXiv preprint arXiv:2508.02062*, 2025.

Chao Tang, Anxing Xiao, Yuhong Deng, Tianrun Hu, Wenlong Dong, Hanbo Zhang, David Hsu, and Hong Zhang. Functo: Function-centric one-shot imitation learning for tool manipulation. *arXiv preprint arXiv:2502.11744*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Vitalis Vosylius and Edward Johns. Instant Policy: In-context imitation learning via graph diffusion. *arXiv preprint arXiv:2411.12633*, 2024.

Shengjie Wang, Jiacheng You, Yihang Hu, Jiongye Li, and Yang Gao. SKIL: Semantic keypoint imitation learning for generalizable data-efficient manipulation. *arXiv preprint arXiv:2501.14400*, 2025.

Yixuan Wang, Guang Yin, Binghao Huang, Tarik Kelestemur, Jiuguang Wang, and Yunzhu Li. GenDP: 3D semantic fields for category-level generalizable diffusion policy. In *8th Annual Conference on Robot Learning*, volume 2, 2024.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. arXiv preprint arXiv:2401.00025, 2023.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. DemoGen: Synthetic demonstration generation for data-efficient visuomotor policy learning. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.157.

Yida Yin, Zekai Wang, Yuvan Sharma, Dantong Niu, Trevor Darrell, and Roei Herzig. In-context learning enables robot action prediction in LLMs. *arXiv preprint arXiv:2410.12782*, 2024.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable Diffusion complements DINO for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36:45533–45547, 2023.

Xinyu Zhang and Abdeslam Boularias. One-shot imitation learning with invariance matching for robotic manipulation. *arXiv preprint arXiv:2405.13178*, 2024.

Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. Visual in-context learning for large vision-language models. *arXiv preprint arXiv:2402.11574*, 2024.

Yifeng Zhu, Arisrei Lim, Peter Stone, and Yuke Zhu. Vision-based manipulation from single human video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024.

A MatchingDataset Generation Details

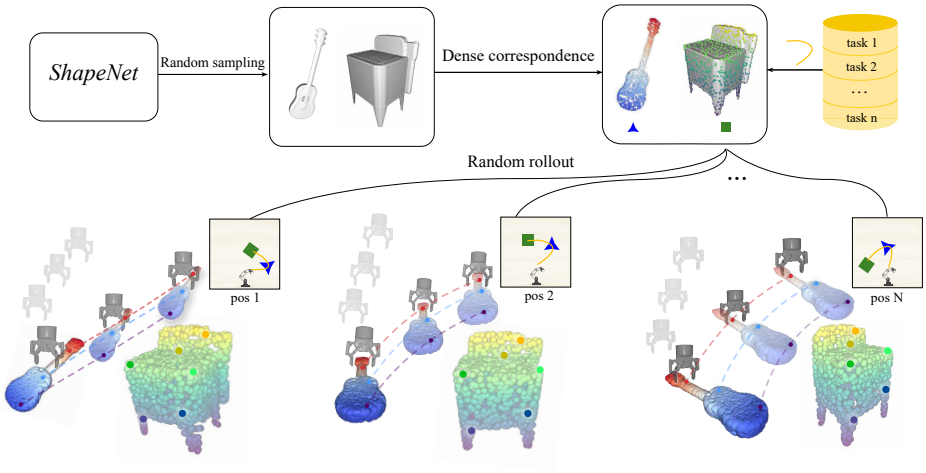

Figure 6: Data generation pipeline of MatchingDataset.

A.1 Dataset Organization

MatchingDataset contains 15K task-consistent trajectory sets and over 150K trajectory realizations, involving more than 50K manipulated-object instances. Each set is initialized from one procedurally constructed reference demonstration. We organize the dataset as

$$\mathcal{D} = \{\mathcal{S}_m\}_{m=1}^{M}, \qquad \mathcal{S}_m = \left\{\tau_m^{(n)}\right\}_{n=1}^{N_m}, \tag{1}$$

where $\mathcal{D}$ denotes the complete dataset, M is the number of task-consistent sets, and m indexes these sets. $\mathcal{S}_m$ is the set initialized from the m-th reference demonstration, N_m is its number of trajectory realizations, and $\tau_m^{(n)}$ denotes its n-th trajectory.

Trajectories within $\mathcal{S}_m$ preserve the same task specification and object models while varying object poses, scene layouts, observations, and robot motions. Each trajectory contains synchronized RGB-D observations, point clouds, robot and gripper states, and end-effector actions, together with ground-truth 3D correspondence and visibility annotations.

A.2 Reference Synthesis and Spatial Retargeting

As shown in Figure 6, we construct reference demonstrations using task-relevant object meshes sampled from ShapeNet (Chang et al., 2015). Each demonstration is defined by an object-centric task specification describing the roles of the involved objects, their interaction regions, and the desired spatial relation or motion. These specifications instantiate common behaviors such as grasping, pick-and-place, pushing or pulling, and opening or closing, as well as lower-frequency behaviors including insertion, rotation, pressing, stacking, hanging, and sorting.

We convert each task specification into object-centric end-effector waypoints and corresponding gripper states, which are interpolated to produce a continuous trajectory. For prehensile interactions, the manipulated object maintains a fixed relative transform to the gripper after grasping. The resulting motion is synthesized kinematically and rendered from the configured camera views to obtain synchronized observations and actions.

Given a reference demonstration, MatchingEngine generates additional realizations through object-centric spatial retargeting. Similar object-centric retargeting principles are used in MimicGen (Mandlekar et al., 2023) and DemoGen (Xue et al., 2025). Specifically, we sample new poses for task-relevant objects and perturb the surrounding scene layout. Because each waypoint is expressed

in the coordinate frame of its associated object, it can be transformed together with the object while preserving the corresponding interaction region.

For multi-object tasks, different trajectory segments may be anchored to different objects. For example, the grasp segment of a pick-and-place trajectory is anchored to the source object, whereas the placement segment is anchored to the target object. We transform each segment using its associated object pose and connect the resulting segments through interpolated free-space motion. The reference demonstration and its retargeted realizations jointly form one task-consistent set $\mathcal{S}_m$.

A.3 Ground-Truth Correspondence Annotation

All trajectories within a task-consistent set retain the same object meshes and local object coordinates. This shared object space allows canonical surface points to define correspondence identities across frames and trajectory realizations.

For clarity, we consider one task-relevant object and omit the set and object indices. We uniformly sample $W = 2048$ canonical surface anchors $\{\mathbf{x}_j\}_{j=1}^{W}$ from its mesh. The world-space position of anchor $\mathbf{x}_j$ in frame t of trajectory n is

$$\mathbf{p}_{j,t}^{(n)} = \mathbf{R}_t^{(n)}\mathbf{x}_j + \mathbf{t}_t^{(n)}, \tag{2}$$

where $\mathbf{R}_t^{(n)}$ and $\mathbf{t}_t^{(n)}$ denote the frame-wise object rotation and translation, respectively. Points transformed from the same canonical anchor $\mathbf{x}_j$ therefore share an exact correspondence identity across frames and trajectories.

Let $\mathcal{P}_t^{(n)}$ denote the observed point cloud in the same frame. We associate each transformed anchor with its nearest observed point:

$$\mathbf{q}_{j,t}^{(n)} = \arg\min_{\mathbf{q}\in\mathcal{P}_t^{(n)}} \left\| \mathbf{q} - \mathbf{p}_{j,t}^{(n)} \right\|_2 . \tag{3}$$

We mark the anchor as visible when

$$\left\| \mathbf{q}_{j,t}^{(n)} - \mathbf{p}_{j,t}^{(n)} \right\|_2 < r_{\text{vis}}, \tag{4}$$

where r_{vis} is the visibility tolerance. Because the point cloud is back-projected from the rendered depth observation, occluded or out-of-view anchors do not have a sufficiently close observed point. For every frame, we store the canonical anchor identity, its associated observed point, and its visibility flag. This provides ground-truth cross-trajectory correspondence supervision without requiring a learned feature matcher during training.

B Real-world Experiment Details and Results

Tasks and Success Criteria. We evaluate four real-world tasks (Figure 7) using the following geometric success criteria: **Pour Water:** the teapot spout is positioned vertically above the cup and the teapot has been tilted by $\geq 15°$; **Put Lid:** the lid covers $\geq 60\%$ of the pot's opening; **Open Box:** the lid has been completely flipped open; **Cut Egg:** the knife has reached the top of the egg yolk and has then moved downward by at least 2 cm.

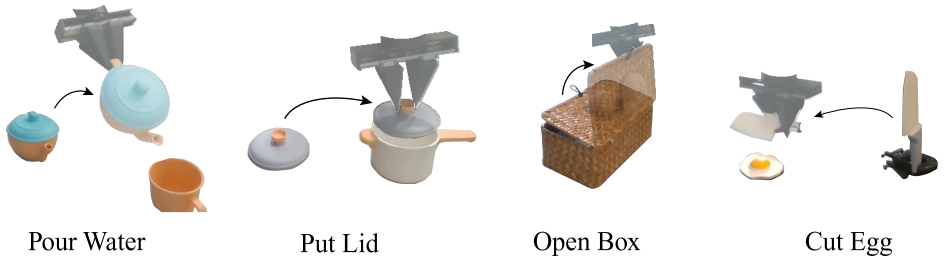

Figure 7: The four real-world tasks used for evaluation.

Three-level Generalization. We assess our models on a set of challenging everyday manipulation tasks, as shown in Figure 7. The object variants are shown in Figure 8. **Layout-Easy:** We randomize the scene by rotating all objects between $-30°$ and $30°$ and applying a random translational offset of 3–5 cm on the table plane; **Layout-Hard:** We apply more aggressive randomization, rotating objects between $90°$ and $150°$ with the same 3–5 cm translational offset; **Shape:** We replace the demonstration objects with semantically similar counterparts that differ in appearance and geometry (Figure 8), and then randomize the scene according to the "Layout-Easy" setting.



Figure 8: Demonstration objects and test variants.

Inter-Category Generalization. For example, given a demonstration of placing a lid onto a pot, our policy can successfully adapt to place a lid onto a novel teapot (see Figure 9).

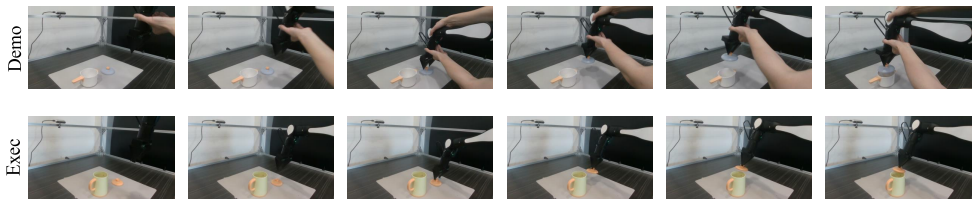

Figure 9: Cross-category generalization.

Interaction Following. To test the nuanced understanding of our policy, we provided two demonstrations of the **Pour Water** task that *differed only in the grasping style*. Our model not only succeeded at the task but also precisely replicated the distinct interaction style from each demonstration (see Figure 10).

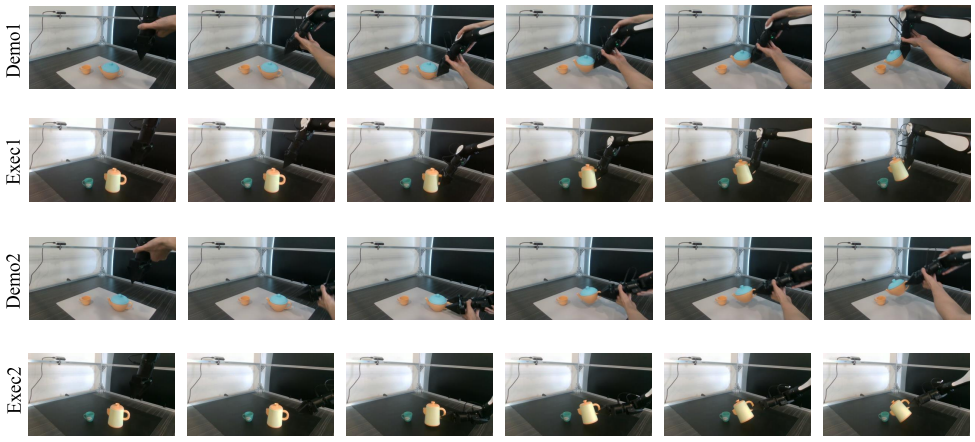

Figure 10: Following the interaction styles in the demonstrations.

C Action Decoding Details

We follow the action representation and rigid denoising procedure of Instant Policy (Vosylius & Johns, 2024). Here k indexes future actions, τ indexes diffusion steps, and 0 denotes the clean target. All relative poses are expressed in the current gripper frame. The formulas below use unnormalized geometric coordinates for clarity.

Geometric targets. Let $q_i \in \mathbb{R}^3$ be canonical gripper keypoint i. A noisy relative pose $\Delta T_{k,\tau} = (R_{k,\tau}, t_{k,\tau})$ places it at

$$p_{k,i,\tau} = R_{k,\tau} q_i + t_{k,\tau}. \tag{5}$$

The translation-induced and rotation-induced denoising targets are

$$\begin{aligned} d^{\text{trans}}_{k,i,\tau} &= t_{k,0} - t_{k,\tau}, \qquad d^{\text{rot}}_{k,i,\tau} = (R_{k,0} - R_{k,\tau}) q_i, \\ p_{k,i,0} - p_{k,i,\tau} &= d^{\text{trans}}_{k,i,\tau} + d^{\text{rot}}_{k,i,\tau}. \end{aligned} \tag{6}$$

The translation target is shared across all gripper keypoints of an action; the rotation-induced target depends on q_i. Concatenating these two 3D components with the scalar gripper-state residual gives the 7D node target. The components describe one shared rigid motion, rather than independent rotations of individual points. We supervise the normalized targets with MSE and undo component normalization before geometric updates.

Noise and rigid pose recovery. Gaussian noise is introduced in the normalized action representation, with poses represented in the Lie algebra of $\mathrm{SE}(3)$. The resulting noisy poses are used to construct gripper nodes, preserving the rigid gripper geometry. Given predicted displacements in geometric coordinates, the clean-position estimate is

$$\widehat{p}_{k,i,0} = p_{k,i,\tau} + \widehat{d}^{\text{trans}}_{k,i,\tau} + \widehat{d}^{\text{rot}}_{k,i,\tau}. \tag{7}$$

A DDIM update (Song et al., 2020) produces candidate next-step points $\widetilde{p}_{k,i,\tau-1}$. We fit a single rigid transformation to the gripper point set by SVD (Arun et al., 1987):

$$\begin{aligned} U_{k,\tau} &= \arg\min_{U \in \mathrm{SE}(3)} \sum_{i=1}^{M_g} \left\| \widetilde{p}_{k,i,\tau-1} - U p_{k,i,\tau} \right\|_2^2, \\ \Delta T_{k,\tau-1} &= U_{k,\tau} \Delta T_{k,\tau}. \end{aligned} \tag{8}$$

Here U acts on homogeneous point coordinates. This alignment enforces a common rigid pose update; the gripper-state variable is denoised separately.

D 3D Lifting and Point Filtering Details

The 3D lifting step is independent of the correspondence source and is applied uniformly to 2D points from semantic matching and visual tracking. We maintain a validity mask for each point: a point is valid only if its depth is available in the RGB-D observation, and tracked points must be predicted as visible by the tracker. This filtering removes points with missing depth, occlusion, or unreliable tracking. Valid 2D points are then back-projected using depth and camera calibration into the robot-aligned world frame. Before policy input, the lifted 3D points at each time step are transformed into the corresponding gripper frame using the inverse gripper pose, yielding temporally consistent 3D point trajectories for downstream manipulation.

E Implementation Details

We trained our model with AdamW at a learning rate of 1×10^{-5}. Training took seven days on four NVIDIA A6000 GPUs for 1M steps. All local evaluations ran on a PC with an RTX 4080 GPU.