HGDPOLICY: HYPERBOLIC GEOMETRY-GUIDED DUAL-SPACE DIFFUSION POLICY FOR FEW-SHOT CONTEXTUAL IMITATION LEARNING IN ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robotic imitation learning aims to rapidly adapt to novel manipulation tasks from only a few human demonstrations. Existing contextual imitation learning methods typically organize demonstrations, observations, and actions as graphs and employ diffusion models for action generation. However, sparse demonstrations require the policy to infer latent subtask relations and transition patterns beyond local state-action correspondences, while Euclidean graph representations are limited in capturing such hierarchical structures. Hyperbolic geometry offers a natural alternative, yet its integration with graph diffusion remains challenging: hierarchical task information must be reliably extracted from few demonstrations and effectively transferred from non-Euclidean representations to anisotropic action generation in Euclidean space. To address these challenges, we propose HGDPolicy, a hyperbolic geometry-guided dual-space diffusion policy for contextual robotic imitation learning. HGDPolicy first introduces transition-aware hyperbolic graph modeling to capture task hierarchies and state-transition patterns from sparse demonstrations. It further develops feasible-action geometric perception, which encodes the learned hyperbolic task structure as an action diffusion prior and dynamically adjusts the center and uncertainty of feasible action distributions according to the current task context. In this way, hierarchical task structure directly modulates the action diffusion process, rather than merely serving as an auxiliary representation. Experiments on nine RLBench generalization tasks show that HGDPolicy achieves a higher average task success rate than the compared baselines, reaching 59.4%, compared with 51.9% for the best-performing baseline. Real-world robotic experiments further demonstrate rapid adaptation to novel tasks with only a few demonstrations and generalization to unseen objects without additional data collection.

1 INTRODUCTION

Robot manipulation aims to enable robots to acquire and execute diverse manipulation skills across tasks and environments. Imitation learning has therefore become a widely used approach for robotic manipulation, as it enables policies to learn task-relevant behaviours directly from expert demonstrations, avoiding the need for manually designed reward functions often required in reinforcement learning. Multi-task imitation learning (Brohan et al., 2022; O'Neill et al., 2024; Team et al., 2024) extends this paradigm by training a shared policy across diverse manipulation tasks, enabling a single model to acquire a broad range of skills. However, this broader task coverage comes at the cost of substantial training and data requirements, often relying on large-scale demonstration datasets and prolonged optimization. Few-shot imitation learning (Duan et al., 2017; James et al., 2018; Bhoopchand et al., 2023) reduces the amount of task-specific data needed for a new task through demonstration-conditioned policies, while meta-learning approaches learn adaptation rules across tasks (Finn et al., 2017). Nevertheless, adapting these policies to a new task can still require task-specific fine-tuning or additional optimization. Contextual, or in-context, imitation learning (Vosylius & Johns, 2023; Di Palo & Johns, 2024; Fu et al., 2025; Vosylius & Johns, 2025) provides an alternative by conditioning a pretrained policy directly on a small set of demonstrations at inference

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.

time, enabling adaptation to new tasks without task-specific parameter updates. This combination of few-shot adaptation and optimization-free deployment has attracted growing interest in contextual imitation learning for robotic manipulation.

Recent methods encode prompts through object relations (Vosylius & Johns, 2023), tokens (Di Palo & Johns, 2024; Vuong et al., 2025) or sensorimotor trajectories (Fu et al., 2025), making context representation integral to skill transfer. Instant Policy (IP) (Vosylius & Johns, 2025) unifies demonstrations, current point-cloud observations and future actions in a graph, casting action prediction as diffusion-based graph generation. Nevertheless, Long-horizon manipulation studies show that complex sequential behaviours are often composed of multiple recombinable subtasks (Fig. 1(a)), where the activation order and temporal coordination of these subtasks determine the execution of the overall behaviour (Triantafyllidis et al., 2023). Therefore, for contextual imitation policies that adapt from a small number of demonstrations, learning local state-action correspondences alone is insufficient; the policy must further infer latent hierarchical relations among subtasks and transition patterns from sparse demonstrations. However, existing Euclidean graph representations are limited in capturing such hierarchical structures (Nickel & Kiela, 2017; Chami et al., 2019) (Fig. 1(c)). Euclidean graph convolutions can incur high distortion when the embedded graph has scale-free or hierarchical structure (Zhang et al., 2021), whereas hyperbolic representations provide an inductive bias for tree-like organization through negative curvature and exponential volume growth (Nickel & Kiela, 2017; Ganea et al., 2018; Chami et al., 2019; Peng et al., 2021). Hyperbolic geometry therefore offers an appropriate candidate for organizing sparse task context (Fig. 1(d)), yet its integration with graph diffusion remains challenging: recent hyperbolic graph diffusion models require a dedicated geometry-aware latent construction rather than a direct substitution of one embedding space for another (Wen et al., 2024). This gives rise to two key challenges: reliably extracting hierarchical task information from few demonstrations and effectively transferring the resulting non-Euclidean representations to guide anisotropic action generation in Euclidean space.

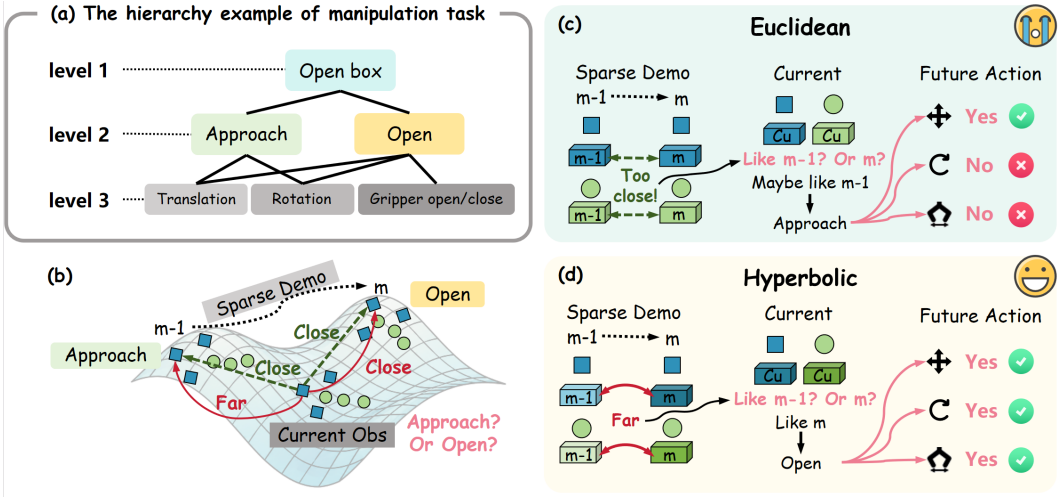


Figure 1: **Motivation of HGDPolicy**. Sparse demonstrations provide limited evidence for latent subtask relations and transition patterns, making local state-action correspondences alone insufficient for action generation.

These two challenges naturally suggest separating hierarchical context modeling from action generation across different geometric spaces. The first challenge calls for a representation that can capture latent hierarchical relations from sparse task context. Hyperbolic geometry is well suited to this role because its negative curvature and exponential volume growth provide an inductive bias for organizing tree-like and hierarchical structures with lower distortion. The second challenge concerns how such non-Euclidean structure can effectively guide action generation in Euclidean space. Directly projecting hyperbolic representations into Euclidean space or replacing them with Euclidean counterparts may weaken the hierarchical structure encoded in the original context. This motivates a dual-space design in which hierarchical task context is modeled in hyperbolic space, while its geometric structure is transferred to Euclidean action space to guide anisotropic pose correction.

Guided by these observations, we propose HGDPolicy, a hyperbolic geometry-guided dual-space diffusion policy for contextual robotic imitation learning. Specifically, to extract hierarchical task information from sparse demonstrations, we introduce a transition-aware hyperbolic graph modeling module. This module organizes demonstration context and current observations in hyperbolic space, capturing temporal task structure and local scene-gripper interactions. It further constructs transition-aware reference states between contextual key steps to characterize latent task hierarchies and stage-transition patterns embedded in sparse demonstrations. To transfer this hierarchical information to Euclidean action generation, we introduce a feasible-action geometric perception module. This module encodes the extracted hyperbolic hierarchy into a diffusion prior for pose correction and models its context-conditioned feasible distribution. By modeling the full covariance of the correction distribution, it preserves coupling across action dimensions and guides diffusion toward anisotropic corrections along task-relevant directions. Future actions are thus generated adaptively under soft geometric constraints inferred from context.

Our contributions are threefold: (1) We introduce a transition-aware hyperbolic graph hierarchy extractor that captures action-specific hierarchical task context and stage-transition patterns across temporal, scene-gripper, and node levels. (2) We develop a feasible-action-geometry-guided anisotropic diffusion process that transfers the extracted hierarchical context to Euclidean action generation by separating systematic correction from full-covariance modelling of conditional pose variation. (3) We conduct extensive evaluations on nine RLBench generalization tasks and real-robot manipulation scenarios, demonstrating improved task generalization, rapid few-shot adaptation, and transfer to unseen objects without additional object-specific data collection.

2 Related Work

Contextual adaptation from demonstrations. Few-shot imitation learning uses demonstrations to condition policies (Duan et al., 2017), infer task embeddings (James et al., 2018) or update parameters through learned adaptation rules (Finn et al., 2017). Contextual imitation learning uses deployment demonstrations as an inference-time prompt, making their representation central to task adaptation. Relational approaches transfer object alignments (Vosylius & Johns, 2023) or task-dependent invariant regions (Zhang & Boularias, 2024), while sequence-based approaches encode sensorimotor prompts (Fu et al., 2025) through keypoint representations (Di Palo & Johns, 2024), action tokenization (Vuong et al., 2025) or state-space models (Yoo et al., 2025). For continuous prediction, demonstration-to-scene correspondences can condition diffusion (She et al., 2026), whereas Instant Policy (IP) (Vosylius & Johns, 2025) jointly represents demonstrations, observations and future actions through graph diffusion. IP provides the starting point for HGDPolicy, which retains its Euclidean nominal action generator within the same demonstration-conditioned setting.

Hierarchical context and hyperbolic representations. Hierarchical manipulation methods differ in what they organize: subtask experts (Triantafyllidis et al., 2023), action representations at multiple spatiotemporal levels (Fateh et al., 2026), or semantically aligned demonstration segments (Yang et al., 2026). These approaches structure skills, actions or demonstration segments; our focus is the hierarchical relations and transition patterns inferred from sparse task context. Hyperbolic embeddings and graph networks provide an inductive bias for hierarchical relations (Nickel & Kiela, 2017; Ganea et al., 2018; Chami et al., 2019), and hyperbolic graph diffusion extends this geometry to latent graph generation (Wen et al., 2024). HGDPolicy applies this representation bias to contextual imitation: transition-aware hyperbolic graph modeling captures temporal task structure and scene-gripper interactions from sparse demonstrations. Its role is to support interpretation of task context, rather than to select a sequence of subtask experts or quantize actions.

Geometry and residual action generation. Conditional denoising provides a generative formulation for visuomotor control (Chi et al., 2025), while action decompositions separate an initial decision from its refinement. Residual reinforcement learning augments a controller with reward-trained corrections (Johannink et al., 2019); PF-DAG (Lei et al., 2026) separates discrete primary-mode selection from mode-conditioned continuous generation. Geometric approaches further structure generation through manifold (De Bortoli et al., 2022) or equivariant models (Ryu et al., 2024; Zhu et al., 2026), or guide actions through spatial value maps (Chen et al., 2024) and learned motion costs (Urain et al., 2023). HGDPolicy combines a continuous nominal action with a demonstration-

conditioned correction. Its feasible-action geometric perception transfers hierarchical task information into a diffusion prior, guiding anisotropic action corrections under soft geometric constraints. This positions the module as a link from task context to action generation; its guidance is a demonstration-induced statistical prior over action corrections, rather than an explicit dynamical constraint enforced by trajectory-projection methods (Bouvier et al., 2025).

3 METHOD

3.1 PROBLEM FORMULATION AND DUAL-SPACE OVERVIEW

Method overview. We consider contextual imitation learning, in which sparse demonstrations and the current observation condition action prediction. This setting requires the policy to infer latent subtask relations and transition patterns from limited context to guide manipulation.

HGDPolicy couples Euclidean action generation with hyperbolic modelling of hierarchical task context (Figure 2). The Euclidean branch retains Instant Policy (IP) as a nominal action generator, providing an anchor for subsequent pose correction. The hyperbolic branch organizes sparse contextual evidence and transfers the captured hierarchical information into a local action distribution in Euclidean pose coordinates. Our method establishes a link between hierarchical context and the distribution governing action correction: hyperbolic representations determine the correction centre and full covariance, allowing task structure to modulate the location and directional variation of residual diffusion. Action generation remains Euclidean, while hyperbolic geometry supplies the hierarchical context that guides its refinement.

The pipeline links hierarchical context to action correction through two modules. First, transition-aware hyperbolic graph modeling module organizes demonstrations and current observations across trajectory-level temporal structures, scene-gripper interaction relations, and node-level geometric representations. Trajectory-level information defines a transition-reference representation of future actions, from which the module captures latent task-relevant information and transition patterns across remaining levels. Second, feasible-action geometric perception module converts the extracted hierarchical information into a correction centre and a full covariance over local pose residuals. The centre captures systematic offsets from the nominal action, while the covariance describes conditional variation and coupling among translation and rotation. Together, these quantities define a prior that constrains anisotropic residual diffusion, with the extracted context also conditioning denoising. The sampled residual is then composed with the nominal action to produce the final end-effector pose sequence.

Problem formulation. Let o_t denote the observation at time t, comprising a scene point cloud, end-effector pose and gripper state, and let a_t denote the corresponding relative end-effector transform and gripper command. The context $\mathcal{D} = \{D_1, \dots, D_K\}$ contains K demonstrations, each a time-ordered sequence of observation-action pairs. Sampled key steps from these demonstrations and the current observation form L Contextual Time Steps, indexed by $\ell \in \{1, \dots, L\}$, for hierarchical context modeling. Conditioned on o_t and $\mathcal{D}$, the policy $\pi_\theta(a_{t:t+H-1} \mid o_t, \mathcal{D})$ predicts a distribution over the next H actions.

The Euclidean IP branch predicts nominal actions $b_{1:H}$, which anchor context-conditioned pose corrections. At horizon position h, the target residual $r_h^\star \in \mathbb{R}^6$ encodes the translation and rotation of $(T_h^b)^{-1} T_h^\star$, where T_h^b and $T_h^\star$ are the nominal and demonstrated end-effector transforms. Residual coordinates are normalized for numerical stability. The predicted residual is converted to a transform and composed with the nominal action, while a separate binary head predicts the gripper state.

3.2 TRANSITION-AWARE HYPERBOLIC GRAPH MODELING (TAHG)

Sparse demonstrations require the policy to infer latent subtask relations and transition patterns from temporal context and local scene-gripper interactions. We organize this information across trajectory-level temporal structures, scene-gripper interaction relations and node-level geometric representations, and construct action-specific transition reference states to guide context extraction.

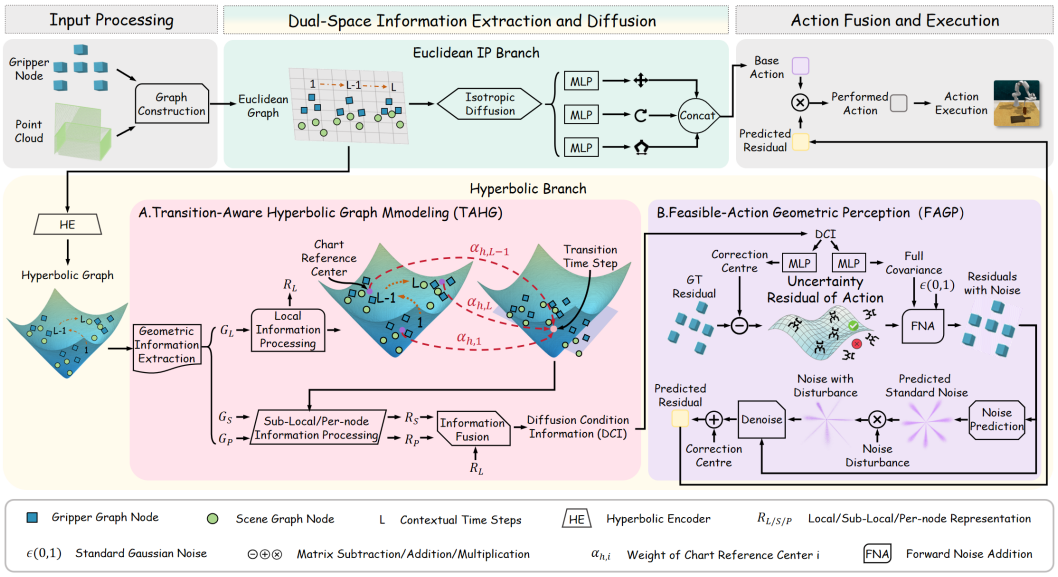


Figure 2: **HGDPolicy pipeline**. The Euclidean branch retains IP as a nominal action generator, while the hyperbolic branch organizes sparse contextual evidence and transfers the captured hierarchical information into a local action distribution in Euclidean pose coordinates.

Organizing hierarchical task context. The context graph contains scene and gripper nodes from the demonstrations and the current observation, with local and cross-observation relations inherited from IP (Vosylius & Johns, 2025). After Euclidean message passing, the Hyperbolic Encoder (HE) maps node features v_i into the d-dimensional Poincaré ball of curvature $-c$ through a learned tangent-space projection g_ψ and the exponential map (Ganea et al., 2018):

$$x_i = \mathrm{Exp}_0^c\big(g_\psi(v_i)\big) \in \mathbb{B}_c^d, \qquad \mathbb{B}_c^d = \{x \in \mathbb{R}^d : c\|x\|^2 < 1\}, \quad c > 0. \tag{1}$$

Each of the L Contextual Time Steps forms a temporal chart, partitioned into scene and gripper subcharts.

For temporal chart ℓ, let $\mathcal{V}_{\ell,k}$ denote the node set of subchart $k \in \{\text{scene}, \text{gripper}\}$. We form subchart reference centers $\mu_{\ell,k}$ and combine them into a chart reference center z_ℓ using Fréchet means (Lou et al., 2020):

$$\begin{aligned} \mu_{\ell,k} &= \arg\min_{\mu} \frac{1}{|\mathcal{V}_{\ell,k}|} \sum_{i \in \mathcal{V}_{\ell,k}} d_c^2(\mu, x_i), \\ z_\ell &= \arg\min_{z} \frac{1}{2} \sum_{k} d_c^2(z, \mu_{\ell,k}), \end{aligned} \tag{2}$$

where d_c is the Poincaré distance. The temporal-chart and scene-gripper subchart levels yield local and sub-local geometric features, G_L and G_S, by fusing tangent-space coordinates of the reference centers and within-group dispersion. Per-node geometric features G_P encode node-to-subchart logarithmic displacements.

Constructing transition-aware Representations. For each future action b_h, a query combines the current local chart and subchart Geometric Features with the nominal action. Multihead attention (Vaswani et al., 2017) over G_L produces the local representation R_L and relevance weights $\alpha_{h,\ell}$ across the L contextual time steps. These weights define the reference state q_h of the transition time step:

$$q_h = \arg\min_{q \in \mathbb{B}_c^d} \sum_{\ell=1}^{L} \alpha_{h,\ell} d_c^2(q, z_\ell), \qquad \alpha_{h,\ell} \geq 0, \quad \sum_{\ell=1}^{L} \alpha_{h,\ell} = 1. \tag{3}$$

The transition reference state q_h guides finer-grained processing by augmenting G_S and G_P with geometric displacements $\mathrm{Log}_{q_h}^c(\mu_{\ell,k})$ and $\mathrm{Log}_{q_h}^c(x_i)$, respectively. Subchart and node-level attention

then produces the sub-local representation R_S and per-node representation R_P. The three representations, R_L, R_S and R_P, are fused into diffusion condition information (DCI), denoted c_h, which integrates temporal task structure with local scene-gripper relations.

3.3 FEASIBLE-ACTION GEOMETRIC PERCEPTION (FAGP)

To transfer hierarchical task information into Euclidean action generation, we encode the extracted context as a diffusion prior for local pose corrections. This prior controls both systematic offsets from the nominal action and directional variation among translation and rotation coordinates.

Predicting the correction centre and full covariance. A shared network maps DCI c_h to a correction centre $m_h \in \mathbb{R}^6$ and a full covariance $\Sigma_h \in \mathbb{R}^{6\times 6}$. The centre captures the context-dependent offset from the nominal action, while the covariance describes the conditional variation and coupling of the six pose coordinates, including translation-rotation coupling. With L_h denoting the Cholesky factor, the centred target residual and covariance satisfy

$$e_h = r_h^\star - m_h, \qquad \Sigma_h = L_h L_h^\top. \tag{4}$$

The covariance is parameterized by a positive-diagonal triangular factor with normalized shape and a learned overall scale; bounded scales and diagonal regularization stabilize its numerical evaluation. Together, the centre and covariance provide a local statistical description of feasible-action geometry that softly constrains residual diffusion.

Guiding anisotropic residual diffusion. We use m_h to centre the target residual and Σ_h to shape its diffusion perturbations. At diffusion level $n \in \{1, \dots, N\}$, let $\bar{\alpha}_n$ be the cumulative signal-retention coefficient of the diffusion schedule (Ho et al., 2020). The forward process is

$$x_{h,n} = \sqrt{\bar{\alpha}_n}\, e_h + \sqrt{1-\bar{\alpha}_n}\, L_h \xi_h, \qquad \xi_h \sim \mathcal{N}(0, I_6). \tag{5}$$

The perturbation covariance (Voleti et al., 2022) is $(1-\bar{\alpha}_n)\Sigma_h$, so the hierarchical context directly modulates the direction and magnitude of residual diffusion around the learned correction centre. The centre and covariance adapt to the current observation and future action but remain fixed within each sampling call.

During denoising, the context-extraction and fusion process incorporates the noisy residual and diffusion level n to produce updated DCI $c_{h,n}^{\mathrm{den}}$. The denoiser predicts standard Gaussian noise from the noisy residual sequence, nominal actions, covariance factors and this updated DCI. At inference, sampling starts from $x_{h,N} = \sqrt{1-\bar{\alpha}_N} L_h \xi_h$ and uses deterministic DDIM updates (Song et al., 2020) with the same covariance to recover a centred residual $\hat{e}_{h,0}$. The correction $\hat{r}_h = m_h + \hat{e}_{h,0}$ is then converted to a residual transform and composed with the nominal action to produce the final end-effector pose sequence.

3.4 LEARNING OBJECTIVE

We supervise the correction centre with an L_1 residual loss to estimate the systematic offset from the nominal action. Gaussian negative log likelihood fits the full covariance to the remaining centred residuals, capturing their scale and cross-coordinate coupling while penalizing excessive covariance volume. The complete objective combines these distribution losses with IP's node-label loss, mean squared error for standard Gaussian noise prediction, L_1 pose reconstruction error and binary cross entropy for gripper classification. For training stability, gradients are stopped through the nominal actions serving as action anchors, the centred residual targets used for covariance estimation, and the covariance factors used for noise injection. Model settings are reported in Appendix B.1; optimizer settings are specified in Appendix A.3; standard Gaussian noise prediction and the Gaussian negative log likelihood are detailed in Appendix B.7.

4 EXPERIMENTS

The experiments were designed to answer three linked questions: whether hyperbolic modelling improves the extraction of hierarchical task context from sparse demonstrations under the same residual-generation design; whether TAHG and FAGP jointly transfer that context into anisotropic

action generation; and whether this pathway supports few-sample adaptation and zero-shot transfer to unseen objects in real-robot evaluations. We address these questions in sequence through task generalization, the two-module comparison, and real-robot tests; parameter analysis is reported in Appendix B.4, and we interpret the results within the evaluated setting.

4.1 Evaluation setting and baselines

Simulation protocol. We evaluated nine generalization tasks from RLBench (James et al., 2020) with a Panda arm and point clouds from front and shoulder cameras. IP (Vosylius & Johns, 2025) and HGDPolicy received two context demonstrations and predicted eight future actions per call. Each task comprised 100 closed-loop rollouts for each evaluation seed, with success determined by the environment's completion signal. All methods were trained once with the same training seed. We evaluated each trained checkpoint at three evaluation seeds; for a fixed task and rollout index, the simulator initialization, including the object pose, was shared across models. We report the mean and standard deviation across the three evaluation seeds and their unweighted task macro-average. Appendix A gives the task, training and evaluation protocol; Appendix B gives the architecture and method configuration.

Baselines. We compared IMOP (Zhang & Boularias, 2024), RVT-2 (Goyal et al., 2024), SAM2Act (Fang et al., 2025), IP (Vosylius & Johns, 2025), and IP with a Euclidean residual branch (IP+ERes). The Euclidean residual tests the choice of representation space for extracting hierarchical context from sparse demonstrations. It retains IP, temporal, scene-gripper and node organization, action-conditioned attention, correction-centre and full-covariance prediction, and residual diffusion. Euclidean features, weighted arithmetic means and vector differences replace Poincaré embeddings, Fréchet means and local logarithmic maps, respectively. RVT-2 and SAM2Act use language conditioning; IP and HGDPolicy use demonstrations. The scores reported are obtained by retraining the methods on the specified fifteen fine-tuning tasks and evaluating them on the designated nine generalization tasks, rather than taken from the original benchmark results reported in the respective baseline papers.

4.2 Multi-task Performance Comparison

We first evaluate HGDPolicy in simulation on nine generalization tasks from RLBench (James et al., 2020), using the same evaluation protocol for all compared methods. The success rates are summarized in Table 1. HGDPolicy achieves the highest task macro-average success rate of 59.4%, compared with 51.9% for IP and 48.5% for IP with a Euclidean residual (IP+ERes). The corresponding gains are 7.6 and 11.0 percentage points, respectively, calculated before rounding. Because IP+ERes and HGDPolicy retain the same nominal IP generator while differing in how contextual information is organized and used for action correction, this comparison isolates the effect of the proposed hyperbolic context modeling and suggests that the improvement is associated with more effective extraction of task stages and stage-transition patterns from sparse demonstrations. The parameter counts are reported in Appendix B.3.

HGDPolicy had the highest mean on nine tasks, while IP had the highest mean on the remaining three. HGDPolicy exceeded the Euclidean residual on all nine tasks, although the margin varied and IP remained stronger on phone placement, plate removal and toilet-seat lowering. This task dependence supports a bounded interpretation: hyperbolic modelling can improve the use of sparse contextual evidence under the evaluated conditions, but it is not uniformly superior to every comparison. Opening the microwave remained difficult for all methods.

To localize the task-level difference within the trajectory, this single-seed stage-wise diagnostic decomposed each rollout into adjacent task-stage transitions under the shared-entry comparison in Fig. 3. HGDPolicy was higher at selected transitions: it more often converted a closure attempt into a target-object grasp on paper roll off, maintained progress from 50% to 75% on toilet seat up, and converted near-completion into the terminal success condition on open box. These contrasts are consistent with more effective extraction and use of hierarchical task context and stage-transition patterns from sparse demonstrations under the same nominal IP generator and residual-generation design. The pattern is task- and transition-dependent: IP remained higher on some transitions, including close box from 25% to 50%, and the toilet-seat-up transition from 95% to Success con-

Table 1: **Success rates (%) on nine RLBench generalization tasks.** IP+ERes denotes IP with Euclidean hierarchical context and residual diffusion.

Model	Phone on Base	Plate Off	Umbrella Out	Paper Roll Off	Open Box	Close Box	Toilet Seat Down	Open Microwave	Toilet Seat Up	Avg.
IMOP	1.0±1.7	0.0±0.0	1.7±0.6	3.3±3.2	1.3±1.5	23.7±4.2	7.7±2.5	0.3±0.6	2.0±1.0	4.6±0.7
RVT-2	1.0±0.0	0.0±0.0	51.0±2.0	13.0±1.0	6.0±1.0	9.3±2.5	57.7±0.6	0.0±0.0	25.3±2.5	18.1±0.7
SAM2Act	0.3±0.6	0.7±1.2	36.3±3.1	16.3±0.6	1.0±1.0	4.3±2.1	52.0±5.6	0.3±0.6	11.0±3.5	13.6±1.1
IP	**94.0±4.4**	**80.3±2.3**	85.3±2.1	20.0±4.0	69.3±3.2	54.7±6.5	**62.3±2.1**	1.0±1.0	0.0±0.0	51.9±2.1
IP+ERes	91.0±5.3	77.7±5.9	91.3±1.2	34.0±5.0	51.3±1.5	49.0±3.6	32.0±3.6	0.0±0.0	10.0±1.7	48.5±1.6
HGDPolicy	92.3±1.5	80.0±2.6	**91.7±1.2**	**34.7±5.5**	**82.0±4.0**	**67.0±2.6**	54.7±4.6	**3.3±1.2**	**29.3±6.8**	**59.4±1.1**

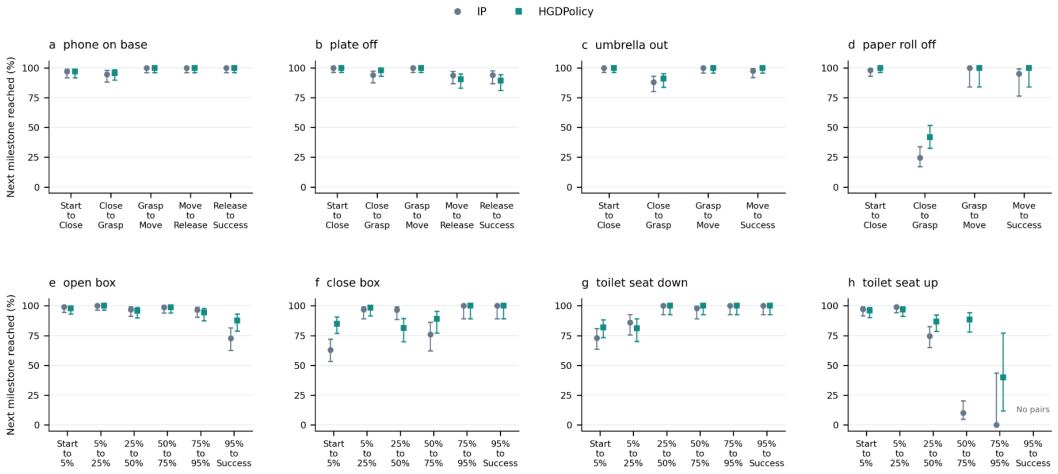


Figure 3: **Task-stage transition analysis**. For each adjacent transition $A \rightarrow B$, the denominator is the set of rollout indices for which both IP and HGDPolicy reached task stage A; markers report the percentage subsequently progressing to task stage B, with Wilson 95% intervals. Object tasks use action/state-defined task stages, whereas joint tasks use normalized progress task stages. "No pairs" denotes zero shared-entry rollouts.

tains no shared-entry pairs. The result therefore provides behaviour-level evidence for improved stage-conditioned use of hierarchical task context within the evaluated setting, rather than proof of an explicit discrete stage variable or uniform superiority across tasks; the task stages are analysis proxies derived from action/state events or normalized progress.

4.3 Ablation Study

We examined TAHG and FAGP by enabling each module separately and together (Table 2). The starting configuration, IP with a simple hyperbolic residual, retains the hyperbolic residual branch with both two modules disabled. It uses basic hyperbolic context representations and isotropic residual diffusion. Enabling graph modeling adds geometric context features and transition-aware reference states; enabling geometric perception adds the learned correction centre and full covariance. Appendix B describes the corresponding module implementations.

As shown in Table 2, the first row corresponds to IP with a simple hyperbolic residual, with both TAHG and FAGP disabled. Starting from 31.7% in task macro-average success rate, enabling TAHG alone increases the average to 44.5%, while enabling FAGP alone yields 39.2%. Enabling both TAHG and FAGP further increases the average to 59.4%, exceeding both single-module configurations as well as IP's 51.9%. These results support the joint contribution of hierarchical context extraction and context-conditioned action-correction geometry.

Table 2: **Ablation study on nine RLBench tasks.** Simple hyperbolic residual retains the hyperbolic branch with both modules disabled.

Modules		Phone on	Plate	Umbrella	Paper Roll	Open	Close	Toilet Seat	Open	Toilet Seat	Avg.
THGM	FAGP	Base	Off	Out	Off	Box	Box	Down	Microwave	Up	
–	–	89.3±1.5	21.3±0.6	84.3±1.5	29.7±4.0	25.7±3.2	12.0±4.4	22.7±3.8	0.0±0.0	0.7±1.2	31.7±1.3
✓	–	93.0±2.6	42.3±4.9	**92.3±1.2**	21.0±6.6	56.0±2.6	23.3±4.9	**54.7±4.2**	0.0±0.0	18.0±2.6	44.5±0.6
–	✓	**95.7±1.5**	62.3±1.5	83.3±4.6	3.7±1.1	13.0±4.0	17.0±1.0	44.7±5.0	0.7±0.6	**32.7±0.6**	39.2±0.5
✓	✓	92.3±1.5	**80.0±2.6**	91.7±1.2	**34.7±5.5**	**82.0±4.0**	**67.0±2.6**	**54.7±4.6**	**3.3±1.2**	29.3±6.8	**59.4±1.1**

4.4 Real-robot adaptation and zero-shot object transfer

Few-sample adaptation. We evaluated six real-robot tasks after fine-tuning. Each training triplet contained two context demonstration trajectories and one supervised query trajectory, following the context/query organisation of IP (Vosylius & Johns, 2025). Three tasks used ten triplets per task and three used one triplet per task; each task was evaluated in ten trials. With ten triplets per task, HGDPolicy succeeded in 8/10, 8/10 and 9/10 trials across the three tasks, compared with 8/10, 6/10 and 5/10 for IP. With one triplet per task, HGDPolicy succeeded in 8/10, 5/10 and 9/10 trials, compared with 5/10, 4/10 and 8/10 for IP (Table 3). The corresponding task macro-averages were 83.3% and 73.3% for HGDPolicy, compared with 63.3% and 56.7% for IP. These panels assess adaptation across tasks with small datasets; the hardware, camera arrangement and staged real-data schedule are specified in Appendix D.

Transfer without new-object demonstrations. We next evaluated opening and closing on one new cabinet and one new box. The cabinet differed in shape and size, and the box differed in size. The adapted policies used demonstrations from the seen objects during fine-tuning as context, without new-object demonstrations or further fine-tuning. As shown in Table 3(c), HGDPolicy succeeded in 7/10, 8/10, 6/10 and 7/10 trials, compared with 4/10, 5/10, 4/10 and 3/10 for IP and 3/10, 2/10, 4/10 and 1/10 for IMOP. The corresponding task macro-averages were 70.0%, 40.0% and 25.0%, respectively. This evaluation tests zero-shot transfer to new objects after task adaptation. Appendix D.4 defines the evaluation groups.

Table 3: **Real-robot success counts.**

(a) 10 triplets per task

Model	Put on Cabinet	Open Box	Open Cabinet	Avg.
IMOP	5/10	2/10	2/10	30.0
IP	**8/10**	6/10	5/10	63.3
HGDPolicy	**8/10**	**8/10**	**9/10**	**83.3**

(b) 1 triplet per task

Model	Close Box	Close Cabinet	Open Fruit Tray	Avg.
IMOP	4/10	3/10	6/10	43.3
IP	5/10	4/10	8/10	56.7
HGDPolicy	**8/10**	**5/10**	**9/10**	**73.3**

(c) Zero-shot transfer to a new box and cabinet

Model	Close Box	Open Box	Close Cabinet	Open Cabinet	Avg.
IMOP	3/10	2/10	4/10	1/10	25.0
IP	4/10	5/10	4/10	3/10	40.0
HGDPolicy	**7/10**	**8/10**	**6/10**	**7/10**	**70.0**

5 Conclusion

We presented HGDPolicy, a dual-space framework that connects hierarchical task reasoning in hyperbolic space with action generation in Euclidean space for few-shot contextual imitation learning. The proposed transition-aware hyperbolic graph modeling extracts latent task structure and stage-transition patterns from sparse demonstrations, while feasible-action geometric perception converts this structure into a context-dependent action geometry that directly shapes anisotropic residual diffusion. This formulation allows hierarchical task information to influence action generation beyond auxiliary feature conditioning. Experiments across simulated and real-world manipulation settings support the effectiveness of the proposed approach for task generalization, few-shot adaptation, and unseen-object transfer.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the method and evaluations. Appendices A, B, C and D provide the simulation protocol, method and implementation details, supplementary diagnostics and real-robot evaluation details.

AI USE STATEMENT

Large Language Models (LLMs) were used to assist with the translation and linguistic refinement of this manuscript, including grammar correction, sentence rephrasing, and improving clarity and readability. The use of LLMs was limited to these language-related tasks and did not replace the authors' scientific judgment or decision-making. All research ideas, methodological choices, experiments, analyses, and conclusions were developed and conducted by the authors. All LLM-assisted text was carefully reviewed and revised by the authors, who take full responsibility for the content, accuracy, and integrity of the manuscript.

REFERENCES

Avishkar Bhoopchand, Bethanie Brownfield, Adrian Collister, Agustin Dal Lago, Ashley Edwards, Richard Everett, Alexandre Fréchette, Yanko Gitahy Oliveira, Edward Hughes, Kory W Mathewson, et al. Learning few-shot imitation as cultural transmission. *Nature Communications*, 14(1): 7536, 2023.

Jean-Baptiste Bouvier, Kanghyun Ryu, Kartik Nagpal, Qiayuan Liao, Koushil Sreenath, and Negar Mehr. Ddat: Diffusion policies enforcing dynamically admissible robot trajectories. *arXiv preprint arXiv:2502.15043*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

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.

Yangtao Chen, Zixuan Chen, Junhui Yin, Jing Huo, Pinzhuo Tian, Jieqi Shi, and Yang Gao. Gravmad: Grounded spatial value maps guided action diffusion for generalized 3d manipulation. *arXiv preprint arXiv:2409.20154*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

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

Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Structured uncertainty prediction networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5477–5485. IEEE, 2018.

Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *Advances in neural information processing systems*, 30, 2017.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. *arXiv preprint arXiv:2501.18564*, 2025.

Fawad Javed Fateh, Ali Shah Ali, Murad Popattia, Usman Nizamani, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. A hierarchical spatiotemporal action tokenizer for in-context imitation learning in robotics. *arXiv preprint arXiv:2604.15215*, 2026.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pp. 357–368. PMLR, 2017.

Max Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, Will Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. Icrt: In-context imitation learning via next-token prediction. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5937–5944. IEEE, 2025.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Stephen James, Michael Bloesch, and Andrew J Davison. Task-embedded control networks for few-shot imitation learning. In *Conference on robot learning*, pp. 783–795. PMLR, 2018.

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.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pp. 6023–6029. IEEE, 2019.

Xiaohan Lei, Min Wang, Wengang Zhou, Xingyu Lu, and Houqiang Li. Primary-fine decoupling for action generation in robotic imitation. *arXiv preprint arXiv:2602.21684*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the fréchet mean. In *International conference on machine learning*, pp. 6393–6403. PMLR, 2020.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80:S1–S7, 2018.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

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: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044, 2021.

Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18007–18018. IEEE, 2024.

Qijin She, Hanyang Yu, Zeming Li, and Ping Tan. Matchingpolicy: Correspondence-aware policy enables cross-object in-context learning. *arXiv preprint arXiv:2608.16715*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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.

Eleftherios Triantafyllidis, Fernando Acero, Zhaocheng Liu, and Zhibin Li. Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network roman. *Nature Machine Intelligence*, 5(9):991–1005, 2023.

Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 5923–5930. IEEE, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vikram Voleti, Christopher Pal, and Adam Oberman. Score-based denoising diffusion with non-isotropic gaussian noise models. *arXiv preprint arXiv:2210.12254*, 2022.

Vitalis Vosylius and Edward Johns. Few-shot in-context imitation learning via implicit graph alignment. *arXiv preprint arXiv:2310.12238*, 2023.

Vitalis Vosylius and Edward Johns. Instant policy: In-context imitation learning via graph diffusion. In *International Conference on Learning Representations*, volume 2025, pp. 35011–35030, 2025.

An Dinh Vuong, Minh Nhat Vu, Dong An, and Ian Reid. Action tokenizer matters in in-context imitation learning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13490–13496. IEEE, 2025.

Lingfeng Wen, Xuan Tang, Mingjie Ouyang, Xiangxiang Shen, Jian Yang, Daxin Zhu, Mingsong Chen, and Xian Wei. Hyperbolic graph diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15823–15831, 2024.

Songhua Yang, Ziyu Liu, Xuetao Li, Ruqi Xiao, Kangxin Zhu, and Miao Li. Ici-vla: In-context imitation with spatiotemporally aligned demonstrations for vision-language-action models. *arXiv preprint arXiv:2609.07581*, 2026.

Youngju Yoo, Jiaheng Hu, Yifeng Zhu, Bo Liu, Qiang Liu, Roberto Martín-Martín, and Peter Stone. Robossm: Scalable in-context imitation learning via state-space models. *arXiv preprint arXiv:2509.19658*, 2025.

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

Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *Proceedings of the web conference 2021*, pp. 1249–1261, 2021.

Xupeng Zhu, Yu Qi, Yizhe Zhu, Robin Walters, and Robert Platt. Equact: An se (3)-equivariant multi-task transformer for 3d robotic manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Simulation Protocol

A.1 Tasks and data

The fifteen-task fine-tuning curriculum comprises tasks from RLBench (James et al., 2020). The tasks are, in order, `basketball in hoop`, `put knife on`, `put umbrella`, `put books on`, `plug charger in`, `unplug charger`, `meat off grill`, `take frame off`, `take lid off`, `open grill`, `open fridge`, `close laptop`, `close microwave`, `turn tap` and `sweep to dustpan`. These tasks define the curriculum and are distinct from the nine generalization tasks reported in the main text. Scenes from ShapeNet (Chang et al., 2015) and RLBench demonstrations are mixed in a 48:48 ratio. Point clouds combine the front, left-shoulder and right-shoulder views, are voxel-processed and represented with 2,048 points per frame.

A.2 Context and rollout configuration

Each training triplet contains two context demonstration trajectories and one supervised query trajectory. The context trajectories are reduced to ten key steps each, while the current observation contributes one step, giving $L = 21$ Contextual Time Steps. The policy predicts $H = 8$ future actions for the query trajectory; key-step selection and the remaining observation/action conventions follow IP wherever the two pipelines share an input contract (Vosylius & Johns, 2025).

A.3 Simulation training and evaluation protocol

The simulation training and evaluation protocol is summarized in Table 4. The staged residual recipe keeps the nominal Euclidean IP branch as the anchor for the residual comparisons.

Table 4: **Simulation training and evaluation settings for the compared methods.**

Item	Setting
Hardware	8 NVIDIA A6000 GPUs
Global batch size	128
Training budget	15,000 steps total across the compared methods
Training seed	A shared training seed 7 for each retraining run
IP recipe	15,000 steps on the fifteen-task curriculum
Staged residual recipe	IP adaptation for 7,500 steps, followed by 7,500 full-model steps
Evaluation seeds	40, 71 and 100
Rollouts	100 closed-loop rollouts per task and seed
Optimizer	AdamW (Loshchilov & Hutter, 2017); learning rate 10^{-5}; weight decay 0.01
Learning-rate schedule	Cosine decay with 500 warm-up steps
Gradient clipping	Gradient norm clipped at 2

A.4 Comparison configurations

The IP with a Euclidean residual branch (IP+ERes) uses the same temporal, scene-gripper and node organization and the same residual diffusion as HGDPolicy, but replaces Fréchet means and local logarithmic maps with weighted arithmetic means and vector differences. The nominal IP action generator and the downstream residual interface are therefore shared by IP+ERes and HGDPolicy; the representation used to extract hierarchical task context is the sole controlled difference.

B Method and Implementation Details

B.1 Architecture and model configuration

The reference architecture and method settings are summarized in Table 5.

Table 5: **Reference architecture and method settings.** The Euclidean and residual diffusion processes use separate samplers; DCI denotes Diffusion Condition Information.

Component	Setting
Graph representation	16 scene nodes per observation; graph hidden dimension 1024; two graph-transformer layers
Hyperbolic representation	Dimension 512; curvature $c = 1$; eight Fréchet iterations
Residual denoiser	Six pose coordinates; hidden dimension 512; four layers; eight attention heads; dropout 0.1
Diffusion	100 training levels; eight DDIM reverse steps per branch; residual $\eta = 0$
Covariance	Full 6×6 covariance for pose residuals; learned log-scale bounded by $[-3, 2]$

Loss terms and weights. For HGDPolicy, the objective combines six terms. The Euclidean node-label loss is the L_1 error between the predicted node-level translation, rotation and gripper labels and Instant Policy-style targets (Vosylius & Johns, 2025). The loss of standard Gaussian noise is the mean squared error between the denoiser's predicted standard noise and the noise sampled in the forward residual-diffusion step. The pose-reconstruction loss is the L_1 error between the decoded continuous six-dimensional pose and the ground-truth continuous pose. The gripper-classification loss is binary cross entropy with logits. When the transition-aware hyperbolic graph modeling module is enabled, the correction-centre loss is the L_1 error between the predicted mean correction and the normalized 6D $SE(3)$ pose residual relative to the detached nominal action. When the feasible-action geometric perception module is enabled, the covariance negative log likelihood evaluates the centred target residual after subtracting the detached mean correction under the predicted full covariance. The total loss is the sum of the active weighted terms.

B.2 Simple hyperbolic residual architecture

The simple hyperbolic residual method retains the Instant Policy Euclidean action backbone and adds a basic hyperbolic context branch. A prefix graph built from the two demonstrations and the current observation is propagated and mapped to the Poincaré ball. The two demonstrations and the current observation form 21 fixed temporal chart corresponding to the 21 Contextual Time Steps. Within each chart, scene and gripper nodes are summarized into subchart reference centers, after which chart, subchart and node-level attention reads the hierarchical context for each future action step. These readouts are fused with the current scene and gripper context, the nominal action and the horizon representation to produce the residual diffusion condition.

B.3 Parameter counts and capacity accounting

The parameter counts of IP, IP with a Euclidean residual, and HGDPolicy are summarized in Table 6. The Euclidean-residual and hyperbolic branches introduce only a small difference in parameter count, whereas HGDPolicy achieves substantially higher performance in the simulation experiments. This performance difference suggests that the gains of HGDPolicy are not primarily attributable to increased model capacity, but may instead arise from its ability to capture hierarchical task-stage relationships and transition patterns in hyperbolic space.

Table 6: **Parameter counts.**

Variant	Total (M)	Residual addition (M)	Updated (M)
Pure Euclidean IP	117.886	0.000	116.531
IP + Euclidean residual	214.838	96.952	96.952
HGDPolicy	215.889	98.003	98.003

B.4 PARAMETER ANALYSIS

We analysed the effects of selected geometric and diffusion parameters, including hyperbolic curvature, the number of contextual time steps, covariance log-scale bounds and the number of hyperbolic residual denoising steps (Table 7). Each parameter group reports the change in success rate relative to its own reference setting. Among the tested configurations, increasing the curvature to $c = 2$ reduced success by 15 percentage points, whereas reducing the number of denoising steps from eight to four changed success by one percentage point. Increasing the number of denoising steps or widening the covariance log-scale range did not consistently improve performance. The reference settings and scope of the reported analysis are given below.

Table 7: **Parameter analysis.** Changes in success rate (percentage points, pp) relative to the corresponding reference setting. Each column pair varies one parameter independently. Bold marks the reference setting; negative values indicate reduced performance relative to the reference.

Model/training settings						Inference	
Curvature c	Δ (pp)	Context L	Δ (pp)	Log-scale	Δ (pp)	Denoising	Δ (pp)
0.5	-5	7	-3	$[-2, 2]$	-2	4	-1
1.0	**0**	15	-8	$[-\mathbf{3}, \mathbf{2}]$	**0**	**8**	**0**
2.0	-15	**21**	**0**	$[-4, 2]$	-6	16	-4

B.5 TRANSITION-AWARE HYPERBOLIC GRAPH MODELING

The Hyperbolic Encoder maps Euclidean graph features into the Poincaré ball (Ganea et al., 2018). For each temporal chart ℓ and subchart k, Fréchet means (Lou et al., 2020) define subchart reference centers $\mu_{\ell,k}$ and chart reference centers z_ℓ. For each future action b_h, multihead attention (Vaswani et al., 2017) over local Geometric Features G_L produces the local Representation R_L and relevance weights. These weights define the reference state q_h of the Transition Time Step, which guides attention over sub-local and per-node Geometric Features G_S and G_P. The three Representations are fused schematically as

$$c_h = \mathrm{MLP}(R_L, R_S, R_P), \tag{6}$$

where R_S and R_P are the sub-local and per-node Representations, and c_h is the DCI supplied to feasible-action geometric perception.

B.6 FEASIBLE-ACTION GEOMETRIC PERCEPTION

FAGP predicts a pose correction centre $m_h \in \mathbb{R}^6$ and a full covariance $\Sigma_h \in \mathbb{R}^{6\times 6}$ from DCI. A positive-diagonal triangular factor contains the 21 free entries of the six-dimensional pose covariance. After the implementation normalizes its shape, we denote the resulting factor by $\widetilde{L}_h$; a scalar log-scale s_h controls its overall magnitude:

$$m_h = \mathrm{MLP}(c_h), \qquad L_h = \exp(s_h)\widetilde{L}_h, \qquad \Sigma_h = L_h L_h^{\mathsf{T}} = \exp(2s_h)\widetilde{L}_h\widetilde{L}_h^{\mathsf{T}}, \qquad \widetilde{L}_{h,ii} > 0. \tag{7}$$

The six pose coordinates are ordered as $(dx, dy, dz, dR_x, dR_y, dR_z)$. The 21 covariance degrees of freedom include translation-rotation and rotation-rotation couplings; the gripper logit is predicted by a separate classification head and does not enter Σ_h. At inference, the denoised residual is added to the predicted centre, $\hat{r}_h = m_h + \hat{e}_{h,0}$, converted to an $SE(3)$ transform and composed with the nominal action.

B.7 STANDARD DDPM-STYLE FORWARD DIFFUSION AND GAUSSIAN NEGATIVE LOG LIKELIHOOD

At horizon position h, we centre the target residual as $e_h = r_h^\star - m_h$ and use the Cholesky factor L_h of the full covariance Σ_h. The standard DDPM-style forward process (Ho et al., 2020) transforms standard Gaussian noise $\xi_h \sim \mathcal{N}(0, I_6)$ through L_h:

$$x_{h,n} = \sqrt{\bar{\alpha}_n}\, e_h + \sqrt{1 - \bar{\alpha}_n}\, L_h \xi_h. \tag{8}$$

Here $x_{h,n}$ is the noisy centred residual at diffusion level $n \in \{1, \ldots, N\}$, with training diffusion steps N and cumulative signal-retention coefficient $\bar{\alpha}_n$. The denoiser predicts $\hat{\xi}_h$, an estimate of the standard Gaussian noise ξ_h, while Σ_h shapes the perturbation (Voleti et al., 2022) and the deterministic DDIM updates (Song et al., 2020).

For the Gaussian likelihood model (Dorta et al., 2018), we express the centred residual as $e_h = L_h z_h$, where $z_h = L_h^{-1} e_h$ is modelled as $z_h \sim \mathcal{N}(0, I_6)$. This auxiliary variable describes residual variation under the likelihood model and is distinct from the injected noise ξ_h. Given DCI c_h, the conditional density $p(e_h \mid c_h)$ of the centred residual is

$$p(e_h \mid c_h) = (2\pi)^{-3} |\Sigma_h|^{-1/2} \exp\left(-\tfrac{1}{2} e_h^\top \Sigma_h^{-1} e_h\right). \tag{9}$$

The Gaussian negative log likelihood for this residual, $\mathcal{L}_{\mathrm{NLL}} = -\log p(e_h \mid c_h)$, is

$$\mathcal{L}_{\mathrm{NLL}} = \frac{1}{2}\left[e_h^\top \Sigma_h^{-1} e_h + \log|\Sigma_h| + 6\log(2\pi)\right]. \tag{10}$$

The quadratic term is the squared Mahalanobis distance (Mahalanobis, 2018), while $\log|\Sigma_h|$ penalizes excessive covariance volume. The complete objective combines the six loss terms described in Appendix B.1.

For training stability, gradients are stopped through the nominal action anchors and the centred residual targets used for covariance estimation. Gradients are also stopped through the covariance factors used for noise injection. Together, the correction centre and covariance provide a local statistical description of feasible-action geometry that softly constrains residual diffusion.

C Additional Covariance Diagnostics

The following diagnostics examine how HGDPolicy allocates pose corrections and conditional variation across tasks. Together, they show how the correction centre and full covariance transfer contextual information into an anisotropic residual distribution.

Task-dependent correction profiles. Figure 4 shows the correction centre, predicted covariance diagonal and action correction relative to the nominal action, from left to right. Each panel averages its respective quantity over the task's predictions and horizon positions. The first and third panels use component-wise absolute values, with action corrections calculated as differences between the final and nominal action coordinates. The correction centre and action correction have larger components in rotational coordinates. The predicted covariance diagonal varies across tasks and across translation and rotation coordinates. These profiles show how HGDPolicy adapts both systematic offsets and directional variation to task context.

Cross-coordinate covariance. Figure 5 shows cross-coordinate coupling separately for each of the nine tasks. For each task, we average the predicted full covariance over predictions and horizon positions, then convert this average to a correlation matrix. Each off-diagonal entry is divided by the square root of the product of the two corresponding diagonal entries. The heatmaps show the absolute values of these correlations on a shared colour scale. Translation-rotation and rotation-rotation coupling recur across tasks, with dR_y-dR_z coupling ranging from 0.061 to 0.068. The pattern also varies with task context; for example, close box shows stronger dx-dz coupling than the other tasks. These structures illustrate how the full covariance represents coordinated variation among the six pose coordinates, supporting anisotropic residual diffusion.

D Real-robot Setup and Evaluation

D.1 Hardware and scene layout

The real-robot platform uses an ARX_X5 arm with two RGB-D cameras: a D435i mounted on the right side of the arm and a D435 mounted on the left side. The cameras observe the shared workspace and are transformed into the robot base frame using the stored calibration. The arm and both cameras are shown in Fig. 6.

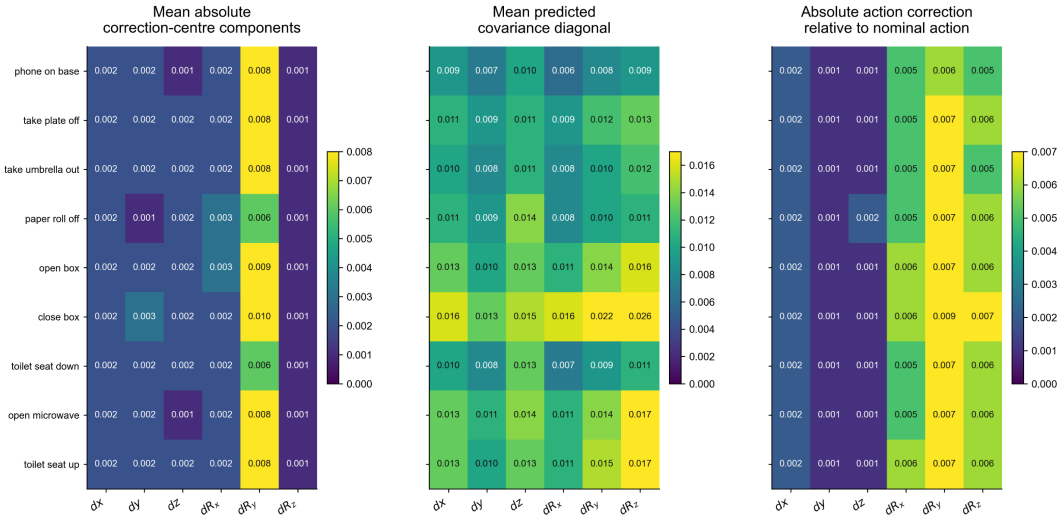


Figure 4: **Task-dependent correction profiles.** Cell annotations report task-level averages over predictions and horizon positions.

D.2 Real-world data collection and processing

Demonstrations are collected by a human operating the arm. Objects are randomly placed within the task workspace before each demonstration, and the resulting context/query organisation follows the simulation format: two context demonstration trajectories and one supervised query trajectory (Vosylius & Johns, 2025). The data are sampled at 10 Hz. Point clouds are transformed to the ARX base frame, clipped by depth and workspace bounds, and filtered to remove end-effector gripper and nearby hand points before sampling. The point-cloud preprocessing pipeline uses a 1-cm voxel scale, statistical outlier removal with 20 neighbours and a 2.0 standard-deviation threshold, followed by random sampling to 2,048 points.

D.3 Real-world Fine-tuning schedule

Real-world adaptation starts from the simulation checkpoint. Data from RLBench (James et al., 2020) and real data are mixed at a 16:48 (1:3) ratio. The model is fine-tuned for 4,500 steps on the real-task set with ten triplets per task, followed by 1,500 steps on three tasks using ten triplets per task and three tasks using one triplet per task. The final checkpoint therefore represents 6,000 real-data adaptation steps after simulation pretraining. The RLBench data use the five tasks included in the simulation fine-tuning, namely `open grill`, `open fridge`, `close laptop`, `close microwave`, and `turn tap`.

D.4 Real-robot Evaluation

Each real task is evaluated over ten trials with randomized object placement. A trial is counted as successful when the corresponding task-specific completion criterion is satisfied: for `Put on Cabinet`, the white block marked with a red "A" is picked up from the table and placed on top of the cabinet; for `Open Box`, the box lid is opened beyond 90°; for `Open Cabinet`, the left cabinet door is opened; for `Close Box`, the box lid is fully closed to 0°; for `Close Cabinet`, the opened left cabinet door is closed; and for `Open Fruit Tray`, the tray lid is opened. The two task-adaptation panels correspond to different task groups: one uses ten training triplets per task, whereas the other uses a single training triplet per task. The unseen-object transfer panel evaluates one cabinet and one box with changed shape or size, using context from the seen objects without additional demonstrations of the new objects or further fine-tuning.

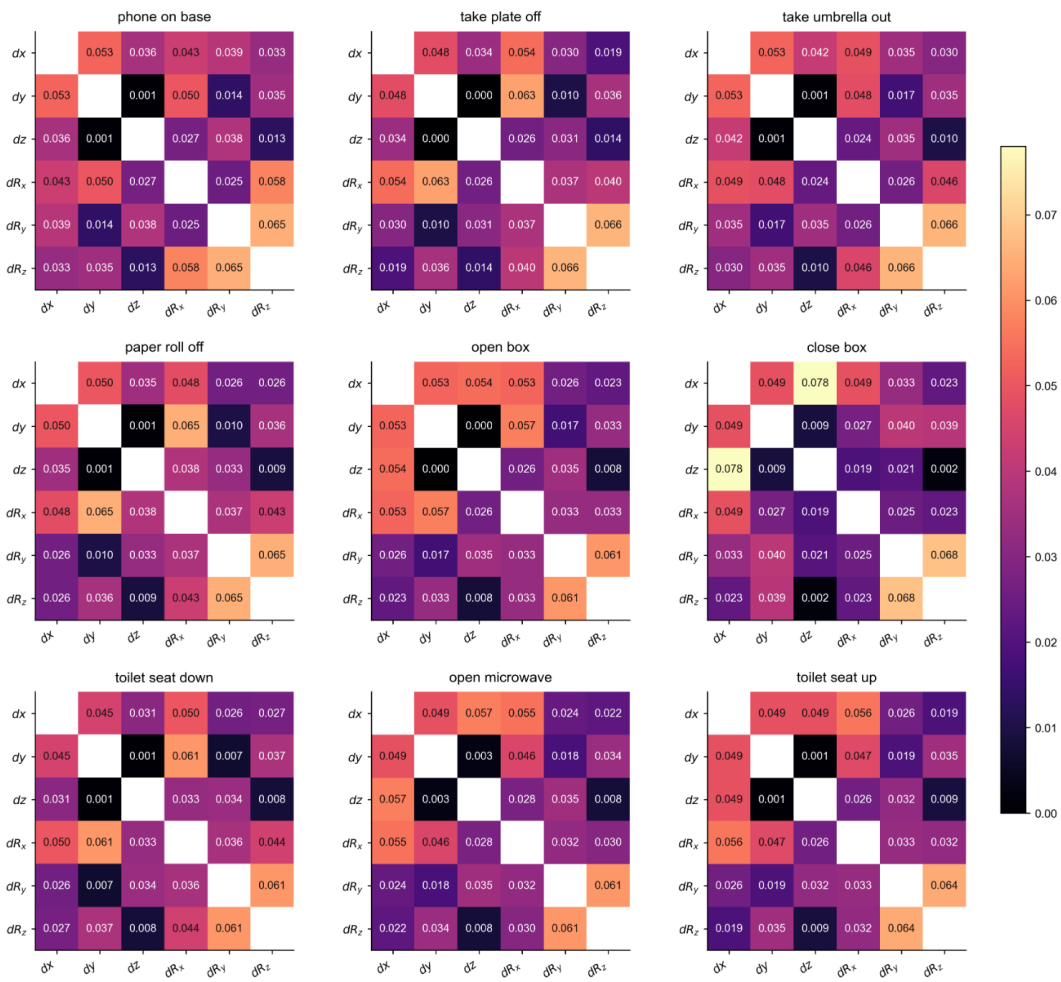


Figure 5: **Cross-coordinate covariance.** Absolute off-diagonal correlations derived from the task-averaged predicted full covariance, shown separately for all nine RLBench generalization tasks. Rows and columns represent the six pose coordinates, and all panels use a shared colour scale. Diagonal entries are omitted, and the gripper coordinate is excluded.

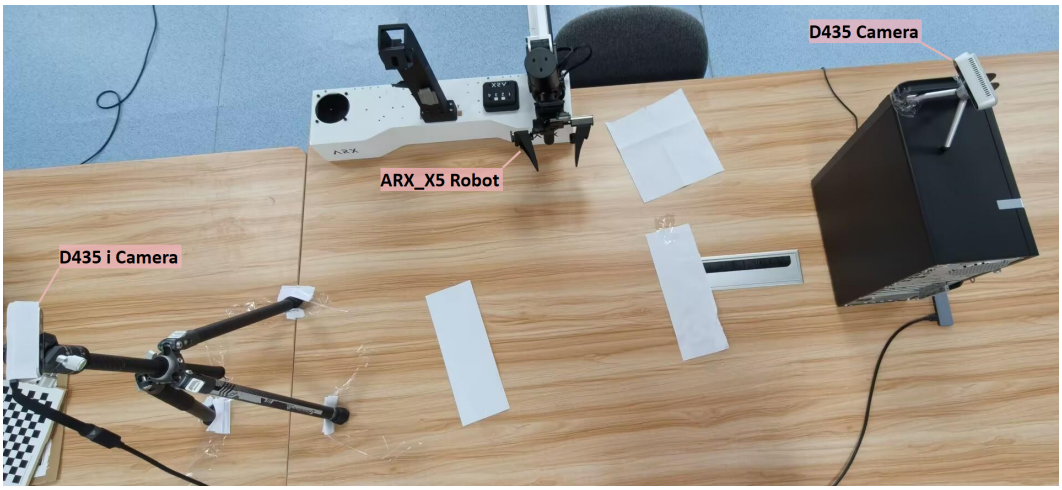


Figure 6: **Real-robot scene and camera placement.** The ARX_X5 arm is shown with the right D435i and left D435 cameras observing the shared workspace.



Figure 7: **Seen and unseen objects used for real-robot adaptation and zero-shot transfer.** The unseen cabinet and box differ in shape or size.

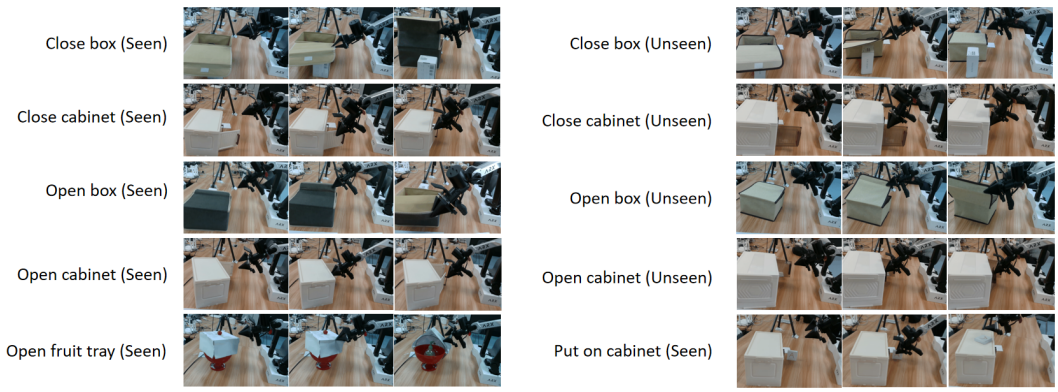


Figure 8: **Real-robot execution examples for seen and unseen objects.**