GeCo-SRT: Geometry-aware Continual Adaptation for Cross-Task Sim-to-Real Transfer

Supplementary Material

1. Implementation and Experimental Analysis

In this section, we provide detailed experimental settings, including simulation training, policy learning with 3D representation, task descriptions, human-in-the-loop data collection, and a qualitative analysis of action adaptation.

1.1. Simulation Training

We use ManiSkill (v3.0.0b21) as the simulator backend. ManiSkill is built on the SAPIEN engine, which utilizes NVIDIA PhysX 5 as the physics engine to provide realistic and precise simulation. The robot models and task environments are adapted from the official ManiSkill benchmark suite, which provides a diverse set of manipulation tasks involving a variety of object types and geometric structures.

1.2. Policy Learning with 3D Representation

Typical RGB observation used in visuomotor policy training suffers from several drawbacks that hinder successful transfer, such as vulnerability to different camera poses and discrepancies between synthetic and real images. To bypass these issues, we propose to use point cloud as the main visual modality. 3D point clouds make it easier for the policy to learn and generalize the geometric knowledge of different objects.

During the simulation training phase, we set up $N = 2$ dual cameras to capture RGBD images. For each camera view, we obtain the raw point cloud $P_{\text{cam}}^{(i)} \in \mathbb{R}^{K\times 3}$ in its local camera frame. We then transform it into the robot's base frame using the camera's known position and orientation:

$$P_{\text{base}}^{(i)} = P_{\text{cam}}^{(i)}(R^{(i)})^T + (p^{(i)})^T \tag{1}$$

Here, $R^{(i)} \in \mathbb{R}^{3\times 3}$ and $p^{(i)} \in \mathbb{R}^{3\times 1}$ denote the i-th camera's orientation and translation in the base frame, respectively. This process is identical to the method used on the real robot (which uses camera calibration), thus aligning the point cloud acquisition method to reduce the sim-to-real gap. After transforming both views to the base coordinate system, the complete scene point cloud P^S is aggregated by concatenating the views:

$$P^S = \bigcup_{i=1}^{N} P_{\text{base}}^{(i)} \tag{2}$$

This aggregated point cloud is then downsampled via Farthest Point Sampling (FPS) and subsequently used as the policy network input.

Figure 1. Real-world workspace setup for human-in-the-loop data collection. The human operator provides online correction through a 3Dconnexion SpaceMouse (marked with a red box) while monitoring the robot's execution.

1.3. Task Description

To verify module transferability, we designed four manipulation tasks of varying difficulty: Pick Cube, Stack Cube, Pick Banana, and Plug Insert. For all tasks, the robot gripper starts at a predefined fixed location, with the tabletop serving as the origin plane. We detail each task's objective, randomized initial conditions, and success criteria below.

- **Pick Cube:** The robot must grasp and lift a cube. The cube is initialized at a random position within a specified range on the tabletop, and the task is successful once the cube is grasped and lifted.
- **Stack Cube:** The robot must grasp the left of two parallel cubes and stack it onto the right one. The two cubes are initialized in parallel with a set gap at random positions. Success requires the target cube to be stably placed on the other cube without falling.
- **Pick Banana:** The robot must grasp and lift a banana-shaped object. The object is initialized at a random position, but its orientation is fixed perpendicular to the gripper's opening. Success is defined as lifting the object without gripping the tablecloth or causing gripper deformation.
- **Plug Insert:** The robot must pick up a plug and insert it into a socket. The plug and socket are initialized at random positions, with the plug always to the left of the socket. Success requires the plug to be steadily inserted into the socket, without gripping the tablecloth or causing gripper deformation.

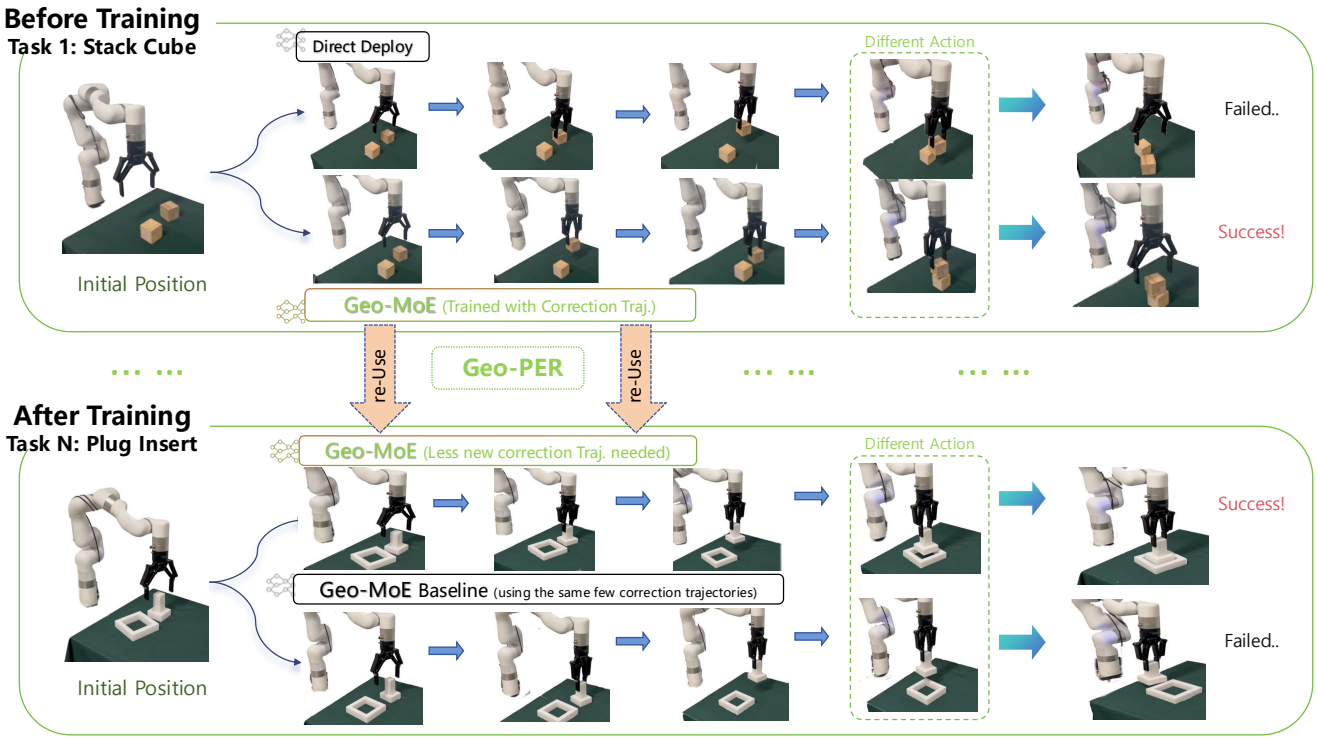

Figure 2. Qualitative visualization of action adaptation on the "Stack Cube" and "Plug Insert" task. **(Top)** In the initial phase (Task 1), human correction trajectories are used to rectify the failure modes of direct deployment. **(Bottom)** In the continual learning phase (Task N), our Geo-MoE model utilizes the *Re-Use* mechanism to achieve success with significantly less new correction data compared to the baseline, demonstrating high data efficiency.

1.4. Human Correction Data Collection

We implement a human-in-the-loop mechanism to harvest high-quality Human Correction Data, and our data collection workspace is shown at 1. The base policy is first deployed to execute the task autonomously. When a policy failure or deviation is observed, the human operator intervenes using a 3Dconnexion SpaceMouse, taking control of the loop to ensure the completion of a successful trajectory. To guaranty the integrity of the training data, a post-processing step is applied: segments corresponding to the policy's erroneous actions prior to the intervention are pruned. This selective filtering ensures that the model updates are based solely on high-quality expert demonstrations, avoiding the negative transfer of failure modes.

1.5. Qualitative Analysis of Action Adaptation

To intuitively understand how our method improves performance, we visualize the action execution trajectories in Figure 2.

Correction for the Sim-to-Real Gap (Task 1). As shown in the top row of Figure 2, the direct deploy policy suffers from domain gaps, generating deviant actions (labeled as "Different Action") that lead to task failure. By introducing human-in-the-loop intervention, our module leverages distinct geometric cues to bridge this gap. This capability allows the model to learn a fine-grained spatial understanding, ensuring the precise action rectification required to successfully complete the "Stack Cube" task.

Efficiency via Knowledge Re-use (Task N). The core advantage is highlighted in the bottom row. Driven by the synergy between Geo-PER and Geo-MoE, our framework transcends simple parameter initialization to actively reuse transferable geometric knowledge. This shared cross-task understanding enables the agent to adapt to Task N with minimal correction data, whereas the baseline fails under the same conditions. This comparison confirms that GeCo-SRT significantly enhances data efficiency by effectively leveraging geometric priors.

1.6. Scalability and Autonomy of HITL

By selectively addressing critical failures, our Human-in-the-Loop (HITL) framework achieves significant performance gains in long-horizon tasks with fewer than 50 human interventions, demonstrating exceptional efficiency and scalability. Furthermore, the framework is agnostic to the correction source; it can seamlessly integrate with MLLM-based agents [2] to facilitate autonomous online error recovery. This transition from human-in-the-loop to model-in-the-loop eliminates manual dependency and further broadens the applicability of our approach in complex, autonomous environments.

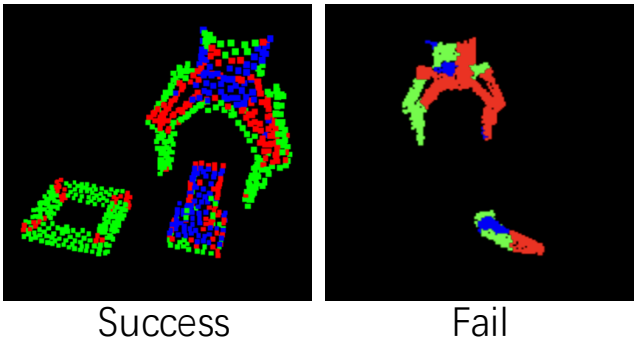

Figure 3. **Qualitative Visualization of Expert Specialization.** Different colors indicate the routing preference of each expert. Our MoE layer naturally learns to partition the input space based on geometric primitives, specializing in edges, corners, and planar surfaces.

1.7. Computational Efficiency and Robustness

Our lightweight point-cloud residual network achieves a real-time inference latency of 26.1ms, competitive with state-of-the-art baselines such as Transic (25.6ms) and Direct Deploy (19.1ms). Our method also exhibits high robustness to hyperparameter variations, particularly regarding the number of experts (Tab. 1).

In terms of memory management, a fixed-size replay buffer retaining 50% of historical data maintains performance parity with full-history training. Specifically, in the Pick-to-Stack Cube task transfer, the N-NBT forgetting rate is 20.0% with a full buffer and only 23.3% with a 50% buffer. When coupled with Geo-PER, this strategy effectively constrains memory growth while ensuring robust knowledge retention.

1.8. MoE Interpretability and Failure Modes

As illustrated in Fig. 3, the routing mechanism exhibits clear interpretability: experts specialize in distinct geometric features. We identify *routing collapse* as a primary failure mode under challenging Out-of-Distribution (OOD) observations (e.g., unseen geometries). In these cases, the gating network disproportionately routes most points to only 1–2 experts, which strongly correlates with subsequent contact failures or grasping instabilities.

Table 1. **Sensitivity Analysis of Expert Numbers (N).** We evaluate the impact of N on Success Rate (SR) and Net Benefit of Transfer (NBT). Performance remains robust across configurations, with $N = 3$ providing an optimal balance between efficiency and specialization.

Number of Experts (N)	Avg. SR (%) ↑	NBT (%) ↓
2	60.0	40.0
3	**66.7**	33.3
8	65.0	**30.0**

Table 2. **Quantitative Results on New Tasks.** Success Rate (SR) for Faucet and Tidying tasks. Our Geo-MoE (continual) significantly outperforms zero-shot and scratch-trained baselines.

Method	Faucet (%) ↑	Tidying (%) ↑
Direct Deploy	10.0	0.0
Geo-MoE (zero-shot)	53.3	30.0
Geo-MoE (scratch)	76.6	43.3
Geo-MoE (continual)	**83.3**	**56.7**

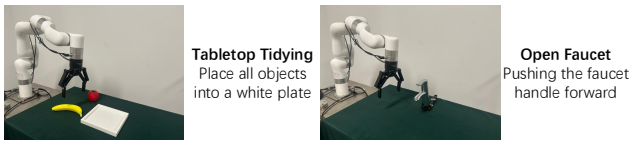

Figure 4. **Visualization of Newly Integrated Tasks.** Experimental setups for *Faucet* and *Tidying* scenarios. Our method effectively handles these novel geometries through efficient knowledge transfer.

1.9. Task Complexity and Diversity

To further evaluate the diversity of our approach, we introduce three challenging scenarios: *Open Faucet* (nonlinear rotation), *Tabletop Tidying* (long-horizon), and *Pick Clear Bottle* (transparency) (Figs. 4, 5). Tab. 2 compares our method against Direct Deploy. Pretrained on the four tasks in the main text, our Geo-MoE module achieves strong zero-shot performance without real-world interaction data. Furthermore, the continual learning setting significantly yields higher success rates than training from scratch, confirming that GeCo-SRT effectively accumulates transferable knowledge for data-efficient adaptation.

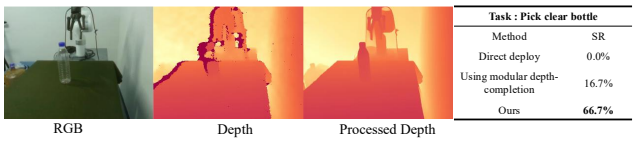

Figure 5. **Qualitative Results of Depth Completion.** Our method effectively restores missing depth information in geometrically complex regions, providing a dense, noise-resilient point cloud for downstream manipulation.

1.10. Handling Depth Invalidity

Our core contribution models sim-to-real as a continual learning problem anchored on geometric features; sensor-level limitations are orthogonal to this framework. However, to demonstrate extensibility, we integrate a modular

depth-completion step [1]. By recovering metric depth from RGB-D inputs (Fig. 5), we ensure sufficient geometric fidelity, enabling Geo-MoE to maintain robust transfer even on challenging surfaces like clear bottles.

1.11. Evaluation with RGB Inputs

We further evaluate GeCo-SRT using RGB-only inputs (Tab. 3). While RGB introduces larger sim-to-real gaps, our geometry-aware experts operate on image patches to achieve reasonable performance. This demonstrates that our framework generalizes to other modalities, although point clouds remain superior due to their direct representation of the geometric invariance that motivates our design.

Table 3. **Success Rate (SR) with RGB Input.** Comparison under RGB-only observations. $T1^{\dagger}$ and $T2^{\dagger}$ denote test scenarios with increased geometric complexity. * indicates results under partial observation.

Method	$T1^{\dagger}$ (%) ↑	$T2^{\dagger}$ (%) ↑
Direct Deploy	3.3	0.0
Domain Randomization	10.0	6.7
Ours	**40.0**	**20/33***

References

[1] Liu et al. Manipulation as in simulation. *ICLR*, 2026. 4

[2] Wenke Xia, Yichu Yang, Hongtao Wu, Xiao Ma, Tao Kong, and Di Hu. Human-assisted robotic policy refinement via action preference optimization. *arXiv preprint arXiv:2506.07127*, 2025. 2