Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

SID: Sliding into Distribution for Robust Few-Demonstration Manipulation

Yicheng Ma[1,2*], Wei Yu[1,2*], Zhian Su[1,2], Xidan Zhang[1,2], Huixu Dong[1,2†]
[1]Grasp Lab, Zhejiang University [2]Torch Kernel Co., Ltd.
[*]Equal contribution [†]Corresponding author, e-mail: huixudong@zju.edu.cn

***Abstract*—Generalizing robotic manipulation across object poses, viewpoints, and dynamic disturbances is difficult, especially with only a few demonstrations. End-to-end visuomotor policies are expressive but data-hungry, while planning and optimization satisfy explicit constraints but do not directly capture the interaction strategies demonstrated by humans. We propose *Sliding into Distribution* (SID), a structured framework that learns an object-centric motion field from canonicalized demonstrations to iteratively *slide* the system toward the demonstrated manifold and into the reliable operating region of a lightweight egocentric execution policy, mitigating out-of-distribution (OOD) execution. The motion field provides large corrective motions when far from the demonstration manifold and naturally vanishes near convergence, enabling robust reaching under substantial pose and viewpoint shifts. Within the reached regime, an egocentric policy trained with conditioned flow matching performs task-specific manipulation, supported by kinematically consistent point-cloud reprojection augmentation that preserves action–observation consistency. Across six real-world tasks, SID achieves approximately 90% success under OOD initializations with only two demonstrations, with under a 10% drop under distractors and external disturbances. Overall, SID provides a new paradigm for few-shot manipulation: explicitly managing distribution shift via online distribution recovery. Project website: https://sliding-into-distribution.github.io/.**

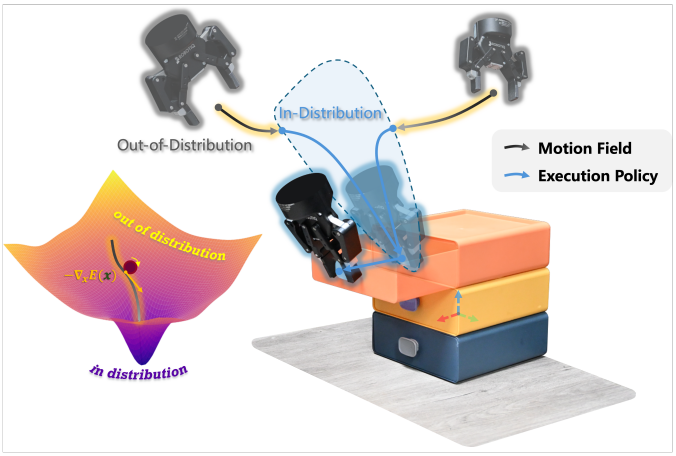

Fig. 1: **SID** combines an object-centric motion field with an egocentric execution policy. From a few demonstrations, the field defines a smooth descent that *slides* OOD states toward the demonstrated support, bringing execution back into the policy's reliable operating region.

I. Introduction

Robust robotic manipulation requires policies that remain reliable under changes in object pose, camera viewpoint, and dynamic disturbances. Achieving such robustness from limited demonstrations is highly desirable in real deployments but remains challenging [8, 47]. In low-coverage datasets, a common failure mode is distribution shift—policies encounter states or observations not supported by the demonstrations—often dominating over architectural expressivity limitations [30, 29]. Moreover, a manipulation episode is rarely homogeneous: it typically contains two qualitatively different regimes (e.g., the approach phase versus execution).

During the approach phase, the robot must recover global geometry to approach a task-relevant target (e.g., a handle or an edge). This stage is often underdetermined, admitting multiple feasible approach trajectories and alignments that are consistent with the demonstrations. During execution, behavior becomes local and interaction-rich, operating in a more constrained regime where it is sensitive to small deviations. When a single end-to-end visuomotor policy [23, 6] is trained to cover both regimes with limited demonstrations, it must simultaneously resolve approach-phase ambiguity and execute precise interaction, making generalization under pose shifts and perturbations brittle—a failure mode frequently observed in offline imitation with low-coverage data [30].

Existing learning-based approaches have made steady progress on generalization and data efficiency, including long-horizon imitation from demonstrations [29, 42] and multi-task policy architectures such as transformer-based visuomotor policies [37]. These methods often treat the episode as a homogeneous control problem. Meanwhile, diffusion-based visuomotor policies [6, 51] can represent complex, multi-modal behaviors, but typically require substantial coverage of approach and interaction variations to remain reliable. Equivariant policy designs [49, 50, 44, 55] often provide strong OOD robustness when the dominant test-time shift aligns with an underlying symmetry (e.g., rigid $\mathrm{SE}(3)$ transforms). That said, enforcing equivariance usually comes with architectural commitments that restrict the hypothesis class and introduce additional design/implementation complexity, potentially trading off some expressive power and ease of deployment for symmetry-consistent generalization. Coarse-to-fine decomposition is particularly effective in one-shot settings [20, 7], suggesting that explicitly structuring the control problem can mitigate OOD failures when data are scarce. However, such coarse-to-fine pipelines are often open-loop at execution time

and can be sensitive to the initial state being close to the first demonstrated waypoint.

Building on this observation, we propose SID, a structured framework that combines (i) an object-centric motion field learned from a small set of canonicalized demonstrations and (ii) an egocentric execution policy. We first canonicalize demonstrations, yielding a consistent motion manifold that suppresses scene- and camera-specific variation. In practice, this canonicalization can be supported by modern 6D pose estimation and tracking [46] and object-level segmentation [2].

From the resulting manifold, we learn a continuous motion field over monotonic approach-phase segments, implemented as a gradient-descent-style dynamical system that attracts states toward the demonstrated support. At test time, integrating the field produces large corrective motions when far from demonstrations and naturally vanishes near convergence, thereby sliding the system back into the reliable operating region of the execution policy—a process we refer to as *sliding into distribution*.

To make the egocentric policy robust under sparse data without corrupting supervision, we introduce kinematically consistent augmentation through point cloud reprojection: we perturb the end-effector pose, reproject segmented wrist-camera point clouds using fixed hand–eye calibration, and update relative actions consistently to preserve action–observation validity. This design is aligned with counterfactual and controllable augmentation paradigms for robot learning [4, 5, 1], while explicitly enforcing geometric consistency. Augmentation is applied only to approach-phase segments, leaving interaction-heavy execution segments unchanged. We also sample a disjoint outer perturbation range to generate OOD observations, which provide negative examples for learning when observations leave the policy's local support.

Finally, within the reached regime, an egocentric execution policy trained via conditioned flow matching performs task-specific manipulation, building on recent progress in flow-based visuomotor policies [53, 15, 38]. Beyond a simple open-loop handoff from the motion field to the policy, SID also supports a closed-loop variant that uses an auxiliary ID-confidence head, trained from augmentation-induced ID/OOD labels, to trigger field-based re-alignment when observations drift from the policy's local support. Experiments show that SID achieves strong spatial generalization and robustness to object displacement and disturbances with as few as one to two demonstrations per task.

Overall, we make the following contributions:

- We propose SID, a structured framework that combines an object-centric motion field for distribution alignment with an egocentric execution policy for task-specific control, and instantiate it with two inference pipelines: an open-loop field-to-policy handoff and a closed-loop re-alignment variant.
- We formulate distribution alignment from few demonstrations as an energy-based dynamical system in object-centric SE(3) space, and learn a smooth motion field that performs gradient-descent-style sliding toward the demonstrated approach manifold.
- We introduce a reprojection-based, stage-aware egocentric augmentation scheme that preserves action–observation consistency under pose perturbations and provides ID/OOD labels for the auxiliary ID-confidence head used in the closed-loop variant.
- We demonstrate robust generalization across poses, disturbances, cluttered scenes, and long-horizon skill compositions on real-world manipulation tasks with only one to two demonstrations per task.

II. Related Work

A. Visuomotor Policy Learning

Visuomotor policy learning trains action-generation policies conditioned on visual observations (e.g., RGB, depth, or point clouds), enabling manipulation with visual feedback [29, 20, 30, 18, 42, 17, 37]. Prior work includes end-to-end image-to-action learning [23, 24], temporally-extended control via sequence modeling [35] or action chunking [56], and diffusion/flow-based policy formulations for multimodal behaviors [6, 51, 53, 38, 15]. Recent methods also incorporate 3D representations to capture geometric structure [51, 21, 43]. Building on these advances, we adopt an egocentric execution policy (trained with conditioned flow matching) and focus on improving its reliability under sparse supervision.

B. Data-Efficient Manipulation Learning

Data-efficient manipulation learning aims to reduce demonstration requirements. Structured policy representations with explicit spatial/geometric inductive biases (e.g., keypoints, transport operators, discretized 3D actions) improve generalization from limited data [52, 36, 37, 45, 31, 12]. Equivariant architectures encode symmetry constraints to improve sample efficiency for pose variations [49, 50, 44, 55]. Other directions include one-/few-shot adaptation via meta-learning [11, 9, 54] and synthetic data/augmentation to expand perceptual and behavioral coverage [57, 48, 4, 5, 1]. Coarse-to-fine decomposition is also effective in one-/few-shot settings by explicitly structuring the control problem [20, 7, 28]. In this spirit, our approach combines a distribution-alignment stage with a task-execution stage to improve data efficiency without increasing the number of demonstrations.

C. Object-centric Policy Learning

Object-centric policy learning improves robustness by representing goals and behaviors relative to objects rather than the global scene. Prior work obtains object-aligned representations via detectors [3, 25, 39], part/affordance segmentation [33, 13, 27, 22], or keypoint-based geometric cues [40, 41, 14, 10, 34, 26], and may further use latent slots [58] or relational abstractions such as scene graphs [19]. These object-centric representations are commonly used to parameterize skills, define subgoals, or condition policies in long-horizon tasks. We similarly leverage object-centric structure, but use it to define and learn a motion field that aligns states back toward the neighborhood of demonstrated trajectories.

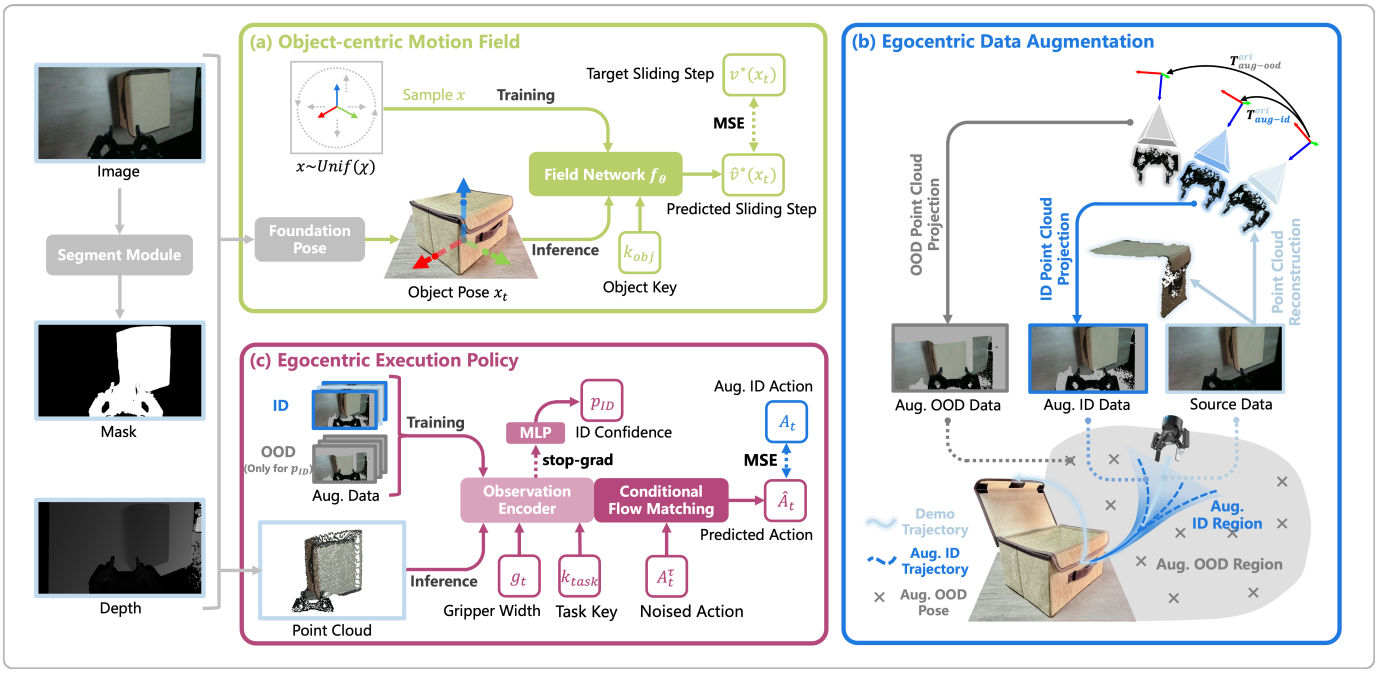

Fig. 2: **SID Overview:** SID comprises three components: (a) an object-centric motion field that predicts sliding steps in object-centric space; (b) an egocentric data augmentation module that generates augmented ID/OOD point-cloud observations via projection and reconstruction; and (c) an egocentric execution policy that predicts actions from point clouds, gripper width, and a task key, with an auxiliary ID-confidence head.

III. Method

We present *Sliding into Distribution* (SID), a structured manipulation framework illustrated in Fig. 2, designed to generalize from extremely few demonstrations across large object-pose variation, changing wrist viewpoints, and dynamic disturbances. SID consists of four components: (i) an *object-centric motion field* f_θ learned from a small set of canonicalized approach demonstrations, (ii) an *egocentric data augmentation* module that expands perceptual diversity via kinematically consistent point-cloud reprojection, (iii) an *egocentric execution policy* π_θ that performs task-specific manipulation once the system is within its reliable operating region, and (iv) two inference pipeline variants: open-loop and closed-loop.

A. Problem Formulation

We study few-demonstration robot manipulation under workspace-wide variation and external disturbances. A task is specified by a small demonstration set $\mathcal{D}_{task} = \{\zeta^{(i)}\}_{i=1}^{M}$ with $M \leq 2$, where each demonstration is a time-indexed sequence $\zeta^{(i)} = \{(o_t^{(i)}, a_t^{(i)})\}_{t=1}^{T_i}$. At time t, the egocentric observation is $o_t = (P_t, g_t)$, where P_t denotes a wrist-centric colored point cloud reconstructed from the wrist-mounted RGB-D camera, and $g_t \in \mathbb{R}$ is the gripper-width signal. The action a_t is a per-step relative end-effector motion between neighboring frames (represented in $\mathrm{SE}(3)$). Our goal is to learn a control system that succeeds under large spatial variation and disturbances while using only $\mathcal{D}_{task}$ as supervision.

B. Object-Centric Motion Field

Why a motion field. A major failure mode of egocentric policies under sparse data is distribution shift: at test time, the robot may start from observations that are far from the demonstrated approach states, causing the policy to operate out of distribution. SID addresses this by learning an object-centric motion field f_θ that funnels the system toward the demonstrated approach manifold (see Fig. 1).

Canonicalization. We use a canonicalization operator $\mathcal{C}$ to extract an object-centric representation x_t and its associated anchor pose $\bar{x}_t$ from an observation:

$$(x_t, \bar{x}_t) = \mathcal{C}(o_t; \xi_t), \qquad x_t \in \mathcal{X},\ \bar{x}_t \in \Omega. \tag{1}$$

Here, ξ_t is a target-object specifier, such as a text prompt. Intuitively, the anchor pose specifies the pose of the target object, and thus anchors an object-centered frame in the egocentric frame. The representation x_t is associated with this anchor, either as an explicit pose or as an object-centric observation, such as segmented RGB or point clouds, associated with the anchor frame.

In general, $\bar{x}_t \in \Omega \subset \mathrm{SE}(3)$ provides the pose used for pose-aligned geometry, and $\mathcal{X}$ denotes the space of object-centric representations anchored by poses in Ω (see Fig. 3). In this work, we use a simple and interpretable instantiation by taking $\mathcal{X} = \Omega$, estimating the target-object pose $T^c_{obj}(t) \in \mathrm{SE}(3)$ in the wrist-camera frame, and setting

$$x_t \triangleq \bar{x}_t \triangleq T^c_{obj}(t) \in \Omega. \tag{2}$$

This pose is used to construct and execute the motion field and to keep the augmentation geometrically consistent, but is not provided as input to the execution policy.

Demonstrated approach manifold. We restrict the motion field to the monotonic approach portion of each demonstration, where a single-valued corrective behavior toward an approach goal is well defined. Let $\mathcal{X}_{\text{demo}} = \{x_j\}_{j=1}^{N} \subset \mathcal{X}$ denote the set of object-centric approach states aggregated from $\mathcal{D}_{\text{task}}$, with corresponding anchor poses $\bar{x}_j \in \Omega$. We use the term *manifold* to refer to the lower-dimensional, task-relevant support formed by the approach-phase states in the demonstrations.

Distance-induced sliding field. We define a pose-aligned potential $E(x)$ using an $\mathrm{SE}(3)$ distance between anchor poses. For notational simplicity, we drop the time index and write $x \in \mathcal{X}$ with anchor pose $\bar{x} = [p, q] \in \Omega$, and similarly $\bar{x}_j = [p_j, q_j] \in \Omega$ for each $x_j \in \mathcal{X}_{\text{demo}}$. We measure similarity with the normalized $\mathrm{SE}(3)$ distance between anchor poses

$$d(x, x_j) \triangleq d_{\mathrm{SE}(3)}(\bar{x}, \bar{x}_j) = \sqrt{\left\| \frac{p - p_j}{\sigma_p} \right\|^2 + \left(\frac{\theta(q, q_j)}{\sigma_r} \right)^2}, \tag{3}$$

where $\theta(q, q_j)$ is the quaternion geodesic angle, and σ_p, σ_r balance translation and rotation.

We want the field to be well defined not only near demonstrations but everywhere in the bounded pose domain Ω. To this end, we sample anchor poses $\bar{x} = [p, q] \sim \mathrm{Unif}(\Omega)$ and instantiate the corresponding input $x \in \mathcal{X}$ (the network input) anchored at $\bar{x}$ (e.g., via reconstruction/rendering; when $\mathcal{X} = \Omega$, we simply take $x = \bar{x}$).

We define a potential induced by the demonstrated set,

$$E(x) \triangleq \tfrac{1}{2}\, d(x, \mathcal{X}_{\text{demo}})^2, \qquad d(x, \mathcal{X}_{\text{demo}}) \triangleq \min_{x_j \in \mathcal{X}_{\text{demo}}} d(x, x_j), \tag{4}$$

and perform a gradient-descent-like update without explicitly computing $-\nabla_x E(x)$. For any query state x, let its nearest demonstrated state be

$$x^*(x) = \arg \min_{x_j \in \mathcal{X}_{\text{demo}}} d(x, x_j). \tag{5}$$

We compute the $\mathrm{SE}(3)$ displacement between anchor poses as a 6D twist

$$\tilde{\Delta}(x \to x^*) = \left[p^* - p,\ \omega(q^*, q) \right]^\top \in \mathbb{R}^6, \tag{6}$$

where $\bar{x}^* = [p^*, q^*] \in \Omega$ is the anchor pose of $x^*(x)$, and $\omega(q^*, q) \in \mathbb{R}^3$ is the axis-angle representation of the relative rotation from q to q^*. We then form a unit descent direction

$$\hat{\Delta}(x \to x^*) = \frac{\tilde{\Delta}(x \to x^*)}{\|\tilde{\Delta}(x \to x^*)\|}, \tag{7}$$

and define the target sliding step

$$v^*(x) = \eta(d(x, x^*(x)))\ \hat{\Delta}(x \to x^*(x)) \in \mathbb{R}^6. \tag{8}$$

Here $\eta(\cdot)$ is a distance-scheduled step size consistent with the quadratic potential $E(x) = \frac{1}{2} d^2$, e.g., $\eta(d) \propto d$ with clipping. Thus, $\hat{\Delta}$ determines the descent direction while $\eta(\cdot)$ determines the magnitude.

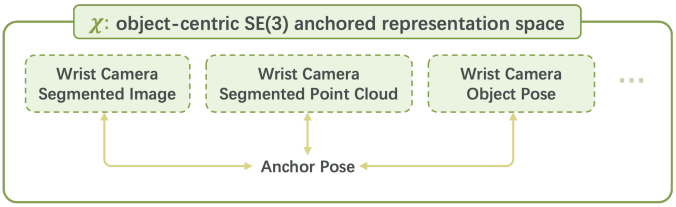

Fig. 3: Illustration of the object-centric representation space $\mathcal{X}$ anchored in $\mathrm{SE}(3)$: different instantiations (e.g., wrist-camera segmented images, segmented point clouds, or explicit object poses) share a common *anchor pose* in $\Omega \subset \mathrm{SE}(3)$, enabling pose-aligned distance computations while keeping the network input in the chosen modality.

Learning and execution. We parameterize the field as $f_\theta : \mathcal{X} \to \mathbb{R}^6$ and regress the target steps

$$\mathcal{L}_{\text{field}} = \mathbb{E}_{x \sim \mathcal{P}(\mathcal{X})} \left[\| f_\theta(x) - v^*(x) \|^2 \right], \tag{9}$$

where $\mathcal{P}(\mathcal{X})$ is the induced sampling mixture obtained by sampling anchor poses $\bar{x} \in \Omega$ and instantiating the corresponding inputs $x \in \mathcal{X}$. At inference time, we obtain the current state x_t, query $v_t = f_\theta(x_t)$, and execute the corresponding end-effector command with step size α. We repeat until $\|f_\theta(x_t)\| < \epsilon_{\text{field}}$, indicating that the system has slid near the demonstrated approach manifold, where ϵ_{field} is a small threshold indicating that the predicted sliding step magnitude is negligible.

Object key for multi-object fields. In practice, the motion field additionally conditions on an object key $k_{obj} \in \mathcal{K}$ that identifies the target object (or instance) within the task, enabling a single network to represent fields for multiple objects:

$$f_\theta : \mathcal{X} \times \mathcal{K} \to \mathbb{R}^6. \tag{10}$$

C. Egocentric Data Augmentation

Egocentric Observation Representation. Following the formulation in Sec. III-A, we use point clouds in the camera frame as the observation, yielding an egocentric representation. This egocentric viewpoint is particularly convenient for data augmentation: we can apply rigid transformations—constrained by the hand–eye calibration—to the non-body component of the observation and the end-effector pose, thereby synthesizing new observations and corresponding end-effector states.

Data Augmentation via Point Cloud Reprojection. To synthesize an egocentric observation under a perturbed viewpoint, we sample a random perturbation from a certain range in the end-effector frame, $T^{ee}_{ee_{\text{aug}}} \in SE(3)$, and convert it to the corresponding camera-frame perturbation using the fixed hand–eye calibration T^{ee}_{c}:

$$T^{c}_{c_{\text{aug}}} = (T^{ee}_{c})^{-1}\, T^{ee}_{ee_{\text{aug}}}\, T^{ee}_{c}, \tag{11}$$

which induces the reprojection transform $T_c^{c_{\text{aug}}} = (T^{c}_{c_{\text{aug}}})^{-1}$ mapping points from the original camera frame to the augmented camera frame.

Crucially, to obtain a physically consistent egocentric observation under viewpoint perturbation, we transform only the target-object point cloud while keeping the gripper point cloud unchanged. We apply a segmentation model to P_t to extract the object point cloud P_t^{obj} and the gripper point cloud P_t^{grip}, and define their union as $P_t^{seg} = P_t^{obj} \cup P_t^{grip}$. For each point $\mathbf{p}_i^c(t) \in P_t^{obj}$, we compute

$$\mathbf{p}_i^{c_{\text{aug}}}(t) = T_c^{c_{\text{aug}}}\, \mathbf{p}_i^c(t), \tag{12}$$

where points are represented in homogeneous coordinates. We leave the gripper points unchanged, i.e.,

$$P_t^{grip,\, c_{\text{aug}}} = P_t^{grip,\, c}, \tag{13}$$

and form the augmented segmented observation by

$$P_t^{seg,\, c_{\text{aug}}} \;=\; P_t^{obj,\, c_{\text{aug}}} \;\cup\; P_t^{grip,\, c_{\text{aug}}}. \tag{14}$$

while keeping the gripper width unchanged, i.e., $g_t^{\text{aug}} = g_t$.

To preserve action–observation consistency under per-frame perturbations, if we sample $T^{ee}_{ee_{\text{aug}}}(t)$ and $T^{ee'}_{ee'_{\text{aug}}}(t{+}1)$ at consecutive time steps, we update the relative action by

$$a_t^{\text{aug}} = \left(T^{ee}_{ee_{\text{aug}}}(t)\right)^{-1} T^{ee}_{ee'}(t \to t{+}1)\, T^{ee'}_{ee'_{\text{aug}}}(t{+}1). \tag{15}$$

where $T^{ee}_{ee'}(t \to t{+}1) = a_t$.

Finally, we control the sampling space of $T^{ee}_{ee_{\text{aug}}}$ to generate both in-distribution and out-of-distribution augmentations. We define ID/OOD relative to the local egocentric support induced by the demonstrations. ID samples are generated from small bounded perturbations around demonstrated end-effector poses and are used, with their geometrically updated actions, to train the execution policy. OOD samples are generated from a disjoint outer perturbation range outside the ID support; they are not used for action-supervised policy training, but serve as negative examples for the auxiliary ID-confidence head. After augmentation, we attach a binary label $p_{\text{ID}} \in \{0, 1\}$ to each sample, where $p_{\text{ID}}{=}1$ indicates ID and $p_{\text{ID}}{=}0$ indicates OOD.

D. Egocentric Execution Policy

Egocentric, Base-Free Policy Interface. To promote generalization, the execution policy is defined in a purely egocentric, base-free manner (no global views or base-frame states are used as inputs). This design decouples execution from absolute workspace placement; a detailed generalization analysis is provided in Appendix B.

It conditions only on $o_t^{seg} = (P_t^{seg}, g_t)$ and predicts an action sequence $A_t = [a_t, a_{t+1}, \ldots, a_{t+H-1}] \in \mathbb{R}^{H \times d_a}$, where H is the action horizon and d_a is the action dimension. Each per-step action a_t comprises (i) a relative end-effector motion in $SE(3)$, (ii) low-dimensional control scalars for the gripper command, and (iii) a termination signal; the rollout stops when this signal exceeds a threshold. To support multiple subtasks with a single model, we additionally condition the policy on a task key k_{task} (embedded as a global conditioning vector).

Conditional Flow Matching over Action Chunks. Following conditional flow matching, we sample a data action chunk

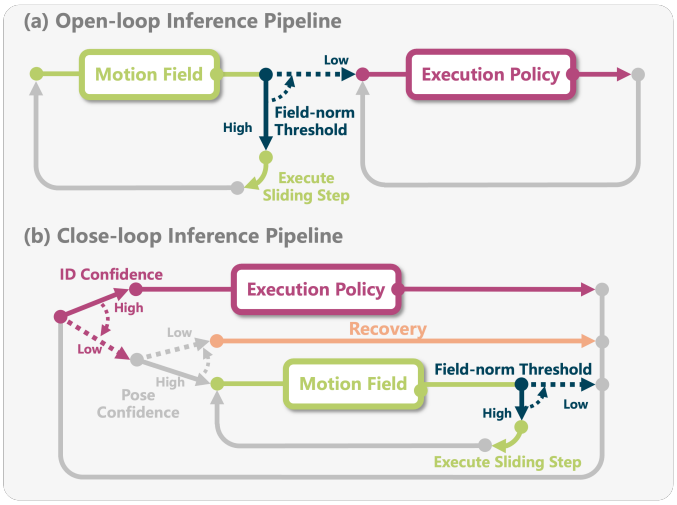

Fig. 4: **Inference pipelines.** (a) Open-loop switches from motion-field sliding (stopped by a field-norm threshold) to policy execution. (b) Closed-loop uses ID/pose confidence to route between policy, motion-field sliding, and recovery.

$A_t \sim p_{\text{data}}(\cdot \mid o_t^{seg}, k_{task})$ and a base sample $A_t^0 \sim p_0(\cdot)$ with $p_0 = \mathcal{N}(0, I)$. We define a linear probability path indexed by the flow time $\tau \in [0, 1]$:

$$A_t^\tau \;=\; (1-\tau)\, A_t^0 \;+\; \tau\, A_t, \qquad \tau \sim \mathcal{U}[0, 1]. \tag{16}$$

The target velocity field along this path is constant:

$$u^*(A_t^\tau \mid A_t, A_t^0) \;=\; \frac{dA_t^\tau}{d\tau} \;=\; A_t - A_t^0. \tag{17}$$

We train a conditional velocity network $u_\theta(A_t^\tau, o_t, k_{task}, \tau)$ by minimizing

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{A_t,\, A_t^0,\, \tau}\left[\left\| u_\theta(A_t^\tau, o_t^{seg}, k_{task}, \tau) - (A_t - A_t^0)\right\|_2^2\right]. \tag{18}$$

At inference time, we sample $A_t^0 \sim \mathcal{N}(0, I)$ and integrate the learned ODE $\frac{dA_t^\tau}{d\tau} = u_\theta(A_t^\tau, o_t^{seg}, k_{task}, \tau)$ from $\tau = 0$ to $\tau = 1$ using Euler updates.

Network Architecture. The velocity network u_θ encodes the egocentric observation using a PointNet-style visuomotor encoder that jointly embeds the segmented point cloud and gripper width into a conditioning feature. Conditioned on this feature, the task embedding, and the flow time, an action head predicts the action-space velocity field over the action chunk.

We additionally attach an auxiliary ID-confidence head c_ϕ to the shared observation encoder. We write its prediction as $p_{\text{ID}} = c_\phi(o_t^{seg}, k_{task})$, denoting an MLP head applied to the encoder feature produced from o_t^{seg} and k_{task}. The output $p_{\text{ID}} \in [0, 1]$ estimates whether the current observation lies within the policy's in-distribution region for the current task/stage. ID samples are used for both action-supervised policy training and ID-confidence prediction, while OOD samples serve only as negative examples for this head. We apply stop-gradient from the auxiliary head to the shared encoder to avoid changing the action-prediction representation.

Algorithm 1 Open-Loop Inference Pipeline

Require: Canonicalization operator $\mathcal{C}$, object alignment anchor ξ_t, motion field f_θ, execution policy π_θ, field-norm threshold ϵ_{field}, termination threshold ϵ_{term}, step size α, object key k_{obj}, task key k_{task}.

Ensure: Task termination

Observe o_t
$x_t \leftarrow \mathcal{C}(o_t; \xi_t)$
while $\|f_\theta(x_t, k_{obj})\| > \epsilon_{field}$ **do** ▷ Motion field
  $v_t \leftarrow f_\theta(x_t, k_{obj})$
  Execute sliding step $\alpha \cdot v_t$
  Observe o_t
  $x_t \leftarrow \mathcal{C}(o_t; \xi_t)$
end while
TERMINATED ← FALSE
while not TERMINATED **do** ▷ Execution Policy
  Observe o_t
  $A_t \leftarrow \pi_\theta(o_t, k_{task})$
  Execute action A_t
  TERMINATED ← (TERMDIM$(A_t) > \epsilon_{\text{term}}$)
end while

Algorithm 2 Closed-Loop Inference Pipeline

Require: Canonicalization operator $\mathcal{C}$, target-object specifier ξ_t, pose score model s, pose confidence threshold ϵ_{pose}, motion field f_θ, field-norm threshold ϵ_{field}, execution policy π_θ, observation encoder with ID confidence head c_ϕ, ID confidence threshold ϵ_{ID}, termination threshold ϵ_{term}, step size α, object key k_{obj}, task key k_{task}.

Ensure: Task termination

TERMINATED ← FALSE
while not TERMINATED **do**
  Observe o_t
  $p_{ID} \leftarrow c_\phi$ ▷ ID confidence
  if $p_{ID} > \epsilon_{ID}$ **then**
    $A_t \leftarrow \pi_\theta(o_t, k_{task})$ ▷ Execution Policy
    Execute action A_t
    TERMINATED ← (TERMDIM$(A_t) > \epsilon_{\text{term}}$)
  else
    $x_t \leftarrow \mathcal{C}(o_t; \xi_t)$
    $p_{pose} \leftarrow s(x_t)$ ▷ Pose confidence
    if $p_{pose} > \epsilon_{pose}$ **then**
      while $\|f_\theta(x_t, k_{obj})\| > \epsilon_{field}$ **do**
        $v_t \leftarrow f_\theta(x_t, k_{obj})$
        Execute sliding step $\alpha \cdot v_t$
        Observe o_t
        $x_t \leftarrow \mathcal{C}(o_t; \xi_t)$
      end while
    else
      Execute RECOVERY action
      continue
    end if
  end if
end while

E. Inference Procedures of SID

SID supports two inference pipelines: an *Open-Loop Inference Pipeline* and a *Closed-Loop Inference Pipeline*. The corresponding procedures are summarized in Alg. 1 and Alg. 2.

Open-Loop Inference Pipeline. As shown in Alg. 1, we first iterate the motion field: the system repeatedly observes o_t, canonicalizes to obtain x_t, queries $v_t = f_\theta(x_t, k_{obj})$, and executes the sliding step $\alpha \cdot v_t$ until $\|f_\theta(x_t, k_{obj})\| \leq \epsilon_{field}$. It then switches to the execution policy, executing action chunks $A_t = \pi_\theta(o_t, k_{task})$ until termination. Termination is determined by the termination signal embedded as one dimension of the d_a-dim action, extracted by TERMDIM(A_t) and thresholded by ϵ_{term}.

Closed-Loop Inference Pipeline. As shown in Alg. 2, the closed-loop pipeline uses the ID confidence p_{ID} to decide whether to execute the policy. If $p_{ID} > \epsilon_{ID}$, the policy action $A_t = \pi_\theta(o_t, k_{task})$ is executed and termination is updated via TERMDIM(A_t). Otherwise, we canonicalize to obtain x_t and evaluate the pose confidence $p_{pose} = s(x_t)$, where $s(\cdot)$ is the score returned by the foundation pose estimator during pose detection. If $p_{pose} > \epsilon_{pose}$, we execute sliding steps from the motion field until $\|f_\theta(x_t, k_{obj})\| \leq \epsilon_{field}$; otherwise, we execute a RECOVERY action and continue the loop. RECOVERY is a hand-designed fallback that moves the end-effector to a predefined safe observation pose, typically a lifted wrist-camera viewpoint with the target workspace in view. It does not perform task manipulation; instead, it is used to reacquire a more reliable segmentation and pose estimate before re-evaluating p_{ID} and p_{pose}.

IV. EXPERIMENTS

We design experiments to answer the following questions.

- **Q1:** Can SID achieve workspace-wide generalization with at most two demonstrations per task, using only augmentation to expand training samples?
- **Q2:** Can SID handle dynamic settings with external disturbances?
- **Q3:** Does SID remain reliable in unstructured, cluttered scenes without carefully curated layouts?
- **Q4:** Can SID scale to long-horizon manipulation tasks that require sustained, multi-step execution?
- **Q5:** Can SID reuse and compose sub-skills across tasks, enabling modular combination of sub-tasks beyond the original demonstrations?

A. Experiment Setups

1) Tasks: To cover the above questions, we evaluate SID on six real-world manipulation tasks: OPEN DRAWER, POUR WATER, HANG CUP, HANG TAPE, PNP-BOX, and MULTI-PNP-BOX. These tasks cover articulated objects, deformable containers, and cluttered scenes, and require both prehensile and non-prehensile manipulation skills, including approaching, grasping, non-prehensile nudging (to open a fabric container), hanging, pouring, and placing.

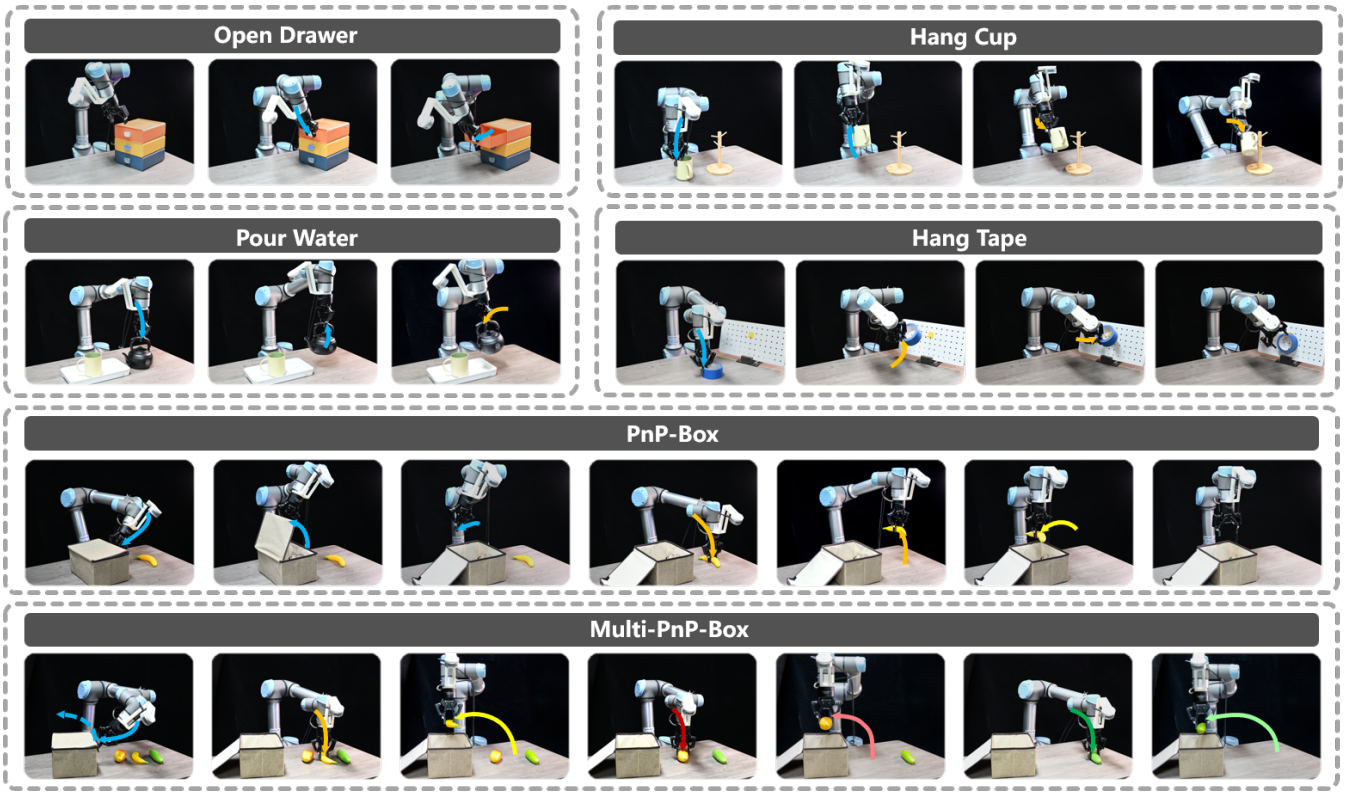

Fig. 5: Visualization of the selected tasks. For each task, the robot motion is decomposed into multiple sub-steps, visualized by trajectories whose colors encode the execution order: blue → orange → yellow → dark red → light red → dark green → light green.

2) Evaluation Protocol: Across all experiments, SID uses only two raw demonstrations per task, which are expanded through data augmentation to form 100 training samples. We provide MT3 and Ret-BC with 10 demonstrations per task, as retrieval-based methods benefit from a larger demonstration memory for alignment and interaction. All remaining trainable baselines are trained with 100 demonstrations per task to ensure stable learning and competitive performance under their standard training regime. We consider five evaluation settings corresponding to Q1–Q5. First, to test *workspace-wide generalization* (Q1), each task is executed from varied initial object poses and camera viewpoints across the reachable workspace, while limiting data collection to at most two human demonstrations per task. Second, to assess *robustness to dynamics and disturbances* (Q2),we apply perturbations to four tasks. For a fair comparison, we evaluate all methods except MT3, Ret-BC, and SID with states constrained to the ID region. Third, to evaluate *unstructured and cluttered scenes* (Q3), we place distractor objects and vary scene layouts without careful curation. Fourth, our task suite includes settings that require *long-horizon, multi-step execution* (Q4). Finally, we include cross-task settings that probe *sub-skill reuse and composition* (Q5) by combining previously seen sub-steps into new task variants. Concretely, we evaluate SID on two novel long-sequence tasks. The first, BOX RETRIEVAL & HANGING, is recomposed from three previously learned skills: PNP-BOX, HANG TAPE, and HANG CUP. The second, PICK NEW OBJECTS INTO BOX, is recomposed from PNP-BOX and PICK OBJECTS (a multi-object picking skill that involves grasping several objects).

3) Baselines: We compare our method against five representative and strong baselines. (1) $\boldsymbol{\pi_{0.5}}$ [16], a vision-language-action (VLA) model trained via co-training on heterogeneous data sources across robot platforms and language-conditioned behaviors to generalize to new environments. (2) **MT3** [7], a fully retrieval-based decomposition method that performs both alignment and interaction via demonstration retrieval at test time. (3) **Ret-BC** [7], a hybrid decomposition baseline that uses retrieval-based alignment followed by behavioral cloning (BC) for interaction. (4) **3D Diffusion Policy (DP3)** [51], a point-cloud diffusion policy that integrates a simplified 3D encoder with robot proprioception for generalizable manipulation. (5) **Action Chunking Transformer (ACT)** [56], a transformer-based policy that predicts temporally extended action chunks from visual observations. All baselines are evaluated under identical experimental conditions for a fair comparison.

4) Metrics: We report **Success Rate** for all tasks. Additionally, we report **Average Length** [32] for long-horizon tasks, which measures execution progress as the number of stages successfully completed. Formally, for a K-stage task, each trial is assigned $l \in \{0, 1, \ldots, K\}$, where l is the index (count)

TABLE I: Results in the static setting under both in-distribution (ID) and out-of-distribution (OOD) evaluations. We report the success rate (%) over 50 trials for each task. *Training Demos* denotes the number of demonstrations used for training. SID-O and SID-C denote SID-open and SID-closed, respectively. † denotes our method.

Methods	Training Demos	Open Drawer		Pour Water		Hang Tape		Hang Cup		PnP-Box		Multi-PnP-Box	
		ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
ACT	100	82	0	68	0	62	0	52	0	56	0	18	0
DP3	100	86	10	74	6	56	2	46	4	72	12	24	0
$\pi_{0.5}$	100	92	14	82	4	84	6	78	6	76	10	64	0
Ret-BC	10	—	56	—	54	—	36	—	62	—	44	—	36
MT3	10	—	76	—	78	—	46	—	74	—	58	—	52
SID-O†	2	—	90	—	86	—	82	—	86	—	84	—	82
SID-C†	2	—	**92**	—	**90**	—	**88**	—	**90**	—	**92**	—	**86**

of the last successfully completed stage; $l = K$ indicates full task completion.

B. Results Comparison

(Q1)SID achieves workspace-wide generalization under OOD initializations with at most two demonstrations per task, consistently outperforming strong baselines. Results are summarized in Table I. Under the OOD setting, all baselines other than MT3 and Ret-BC suffer pronounced performance degradation. By comparison, both SID-open and SID-closed remain robust in OOD, and outperform these baselines not only in OOD but also relative to their ID performance.

We further observe that SID consistently surpasses MT3 and Ret-BC across tasks. We attribute this improvement to SID's ability to update its motion field online using the latest egocentric observations while transitioning from OOD regions toward the ID manifold, enabling more accurate re-entry into the training distribution. In contrast, MT3 and Ret-BC perform trajectory alignment based on object poses estimated from a single first-frame global observation, making them sensitive to pose estimation errors that can propagate to downstream manipulation accuracy. We also observe from Table I that SID-closed consistently outperforms SID-open. We attribute this gain to the fact that SID-closed corrects distribution shifts not only during the approach phase, but also continues to mitigate distribution drift in the post-contact phase, where inaccuracies in policy actions can otherwise accumulate and push the system further away from the training distribution.

Overall, these results provide clear evidence that SID delivers strong workspace-wide generalization with minimal demonstrations: OOD-to-ID motion and augmentation enable effective learning from only two demos per task, and the closed-loop variant further improves robustness by continuously correcting distribution drift throughout execution.

(Q2) SID performs robustly under external disturbances by correcting deviations online. As shown in Table II, $\pi_{0.5}$ exhibits the strongest dynamic robustness across all baselines, which we attribute to its closed-loop hierarchical policy with re-planning from fresh observations and its continuous action-expert design. MT3 and Ret-BC suffer substantial performance drops in dynamic scenarios; both degrade severely when perturbations are present, because retrieval-based methods perform alignment only w.r.t. the first observation frame and do not update the alignment online. Ret-BC performs relatively better, likely because the BC-based execution stage provides additional robustness.

TABLE II: Results in the dynamic setting. We report the success rate (%) over 50 trials per task (aggregated over perturbation cases).

Task	ACT	DP3	$\pi_{0.5}$	Ret-BC	MT3	SID-O†	SID-C†
Hang Tape	6	14	80	12	4	44	**84**
Hang Cup	2	8	68	4	0	36	**88**
PnP-Box	2	10	74	8	2	52	**86**
Pour Water	4	10	78	6	2	40	**82**

SID-open is more robust to *disturbances* since its motion field is continuously updated from the latest observations in approach phases, guiding the robot back toward in-distribution states. However, SID-open can still degrade under stronger disturbances in execution phases: once the disturbance pushes observations outside the in-distribution region, the execution policy alone often fails to recover. In contrast, SID-closed remains robust across both approach and execution phases; its OOD detection module provides a closed-loop safeguard that effectively handles OOD events induced by disturbances.

(Q3) SID remains reliable in unstructured, cluttered scenes. As shown in Table III, MT3, Ret-BC, and both SID variants maintain strong performance under cluttered, unstructured layouts. We attribute this robustness primarily to the use of vision foundation models: SAM2/SAM3 provide explicit target-object segmentation, which filters out background clutter and distractors and yields more stable, target-centric visual inputs for downstream planning and control. As a result, these methods are less sensitive to layout curation and remain reliable even with non-canonical object arrangements. In contrast, the remaining baselines degrade markedly in clutter, since their visual representations are more easily dominated by irrelevant objects and occlusions when explicit target segmentation is unavailable, leading to mis-localization and cascading execution errors.

(Q4) SID can handle long-horizon manipulation tasks that require sustained, multi-step execution. As shown in Table I, SID achieves consistently strong performance on two long-horizon tasks with more than three sequential stages, outperforming the baselines. We attribute this advantage to SID's decomposition design: the motion field facilitates

TABLE III: Results in cluttered environments with added distractor objects. For each task, we report the success rate (%) over 50 trials per task for each method.

Task	ACT	DP3	$\pi_{0.5}$	Ret-BC	MT3	SID-O†	SID-C†
Hang Cup	42	24	72	58	72	86	**88**
PnP-Box	44	28	74	40	54	84	**90**

transitions between the subtask-specific state distributions of consecutive stages, while the execution policy only needs to operate within each subtask-relevant, narrow distribution. This separation substantially reduces the learning burden of the neural policy, enabling reliable multi-stage execution with a relatively compact model. Overall, the results suggest that decomposing long-horizon manipulation into distribution-aware subtask transitions is key to sustained, multi-step control.

(Q5) SID can reuse and compose sub-skills across tasks, enabling modular combination of sub-tasks beyond the original demonstrations. Since both components in SID are lightweight, we can load multiple previously trained SID models when solving a new task and invoke them stage-by-stage, enabling practical skill reuse and re-composition. As shown in Fig. 6, SID achieves competitive performance on both recomposed tasks by reusing pre-trained sub-skill models and invoking them on demand at each stage. This result highlights SID's ability to recombine existing skills to solve novel long-horizon sequences without retraining an end-to-end policy.

Meanwhile, the per-stage success rates in Fig. 6 indicate a consistent gap between *reuse* and the original single-skill success rates. The degradation is most pronounced for `Pick tape` and `Pick cup`, because grasping in the original tasks is performed without occlusions, whereas in the recomposed setting the box introduces additional clutter and partial occlusion that interferes with perception and approach. We also observe reduced performance in the `Place` stages: since the manipulated objects differ from those seen in the original PnP-Box demonstration, the resulting distribution shift further degrades placement quality.

In summary, SID enables effective skill reuse and modular re-composition for new long-horizon tasks, while the remaining performance gap is largely explained by occlusion-induced grasping difficulty and distribution deviations introduced by novel objects and interactions.

V. Conclusions and Limitations

We presented *Sliding into Distribution* (SID), a few-demonstration manipulation framework that mitigates distribution shift by pairing an object-centric motion field for online alignment with an egocentric execution policy trained via conditioned flow matching. SID learns a smooth descent field from canonicalized approach demonstrations to *slide* OOD states back toward the demonstrated manifold, and improves data efficiency through kinematically consistent point-cloud reprojection augmentation. Across real-world tasks, SID shows strong workspace-wide generalization and robustness to disturbances with only one to two demonstrations, while the closed-loop variant further uses an auxiliary ID-confidence signal to monitor policy-side distribution drift and route control back to field-based re-alignment or recovery when needed. SID also supports skill reuse and composition for modular long-horizon manipulation.

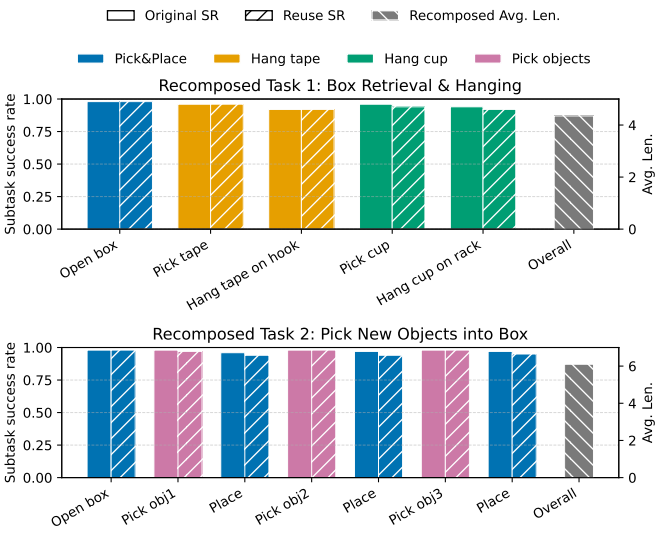

Fig. 6: Results on recomposed tasks. We report the success rate for each sub-skill and the Average Length for the overall recomposed tasks, which counts the number of stages successfully completed.

Limitations: *Pose estimation dependency.* SID relies on an off-the-shelf 6D pose estimator; when pose estimates are unreliable, such as for transparent objects or rapid motion, the motion field can degrade. Future work could reduce this dependency by conditioning on object-centric inputs such as segmented point clouds or RGB. *Limited scene awareness and approach feasibility.* Our method does not explicitly model obstacles or collision constraints, and its motion-field alignment assumes that the demonstrated approach manifold remains observable and reachable. Performance may degrade when feasible approach regions are occluded, blocked, or constrained by cluttered environments. *Limited cross-object generalization.* We do not yet demonstrate strong generalization to unseen objects or categories; more object-agnostic representations are a natural next step.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China Youth Program under Grant No. 52305037, Zhejiang Provincial Natural Science Foundation of China under Grant No. LD26E050001, the "Pioneer" and "Leading Goose" R&D Programs of Zhejiang Province under Grants No. 2025C01072.

References

[1] Ezra Ameperosa, Jeremy A. Collins, Mrinal Jain, and Animesh Garg. Rocoda: Counterfactual data augmen-

tation for data-efficient robot learning from demonstrations, 2025. URL https://arxiv.org/abs/2411.16959.

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[3] Alexandre Chapin, Bruno Machado, Emmanuel Dellandrea, and Liming Chen. Object-centric representations improve policy generalization in robot manipulation. *arXiv preprint arXiv:2505.11563*, 2025.

[4] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation, 2023. URL https://arxiv.org/abs/2302.06671.

[5] Zoey Chen, Zhao Mandi, Homanga Bharadhwaj, Mohit Sharma, Shuran Song, Abhishek Gupta, and Vikash Kumar. Semantically controllable augmentations for generalizable robot learning, 2024. URL https://arxiv.org/abs/2409.00951.

[6] 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, 2024. URL https://arxiv.org/abs/2303.04137.

[7] Kamil Dreczkowski, Pietro Vitiello, Vitalis Vosylius, and Edward Johns. Learning a thousand tasks in a day. *Science Robotics*, 10(108), November 2025. ISSN 2470-9476. doi: 10.1126/scirobotics.adv7594. URL http://dx.doi.org/10.1126/scirobotics.adv7594.

[8] Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets, 2023. URL https://arxiv.org/abs/2304.08742.

[9] Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning, 2017. URL https://arxiv.org/abs/1703.07326.

[10] 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)*, pages 8307–8314. IEEE, 2025.

[11] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning, 2017. URL https://arxiv.org/abs/1709.04905.

[12] Jianfeng Gao, Zhi Tao, Noémie Jaquier, and Tamim Asfour. K-vil: Keypoints-based visual imitation learning. *IEEE Transactions on Robotics*, 39(5):3888–3908, October 2023. ISSN 1941-0468. doi: 10.1109/tro.2023.3286074. URL http://dx.doi.org/10.1109/TRO.2023.3286074.

[13] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2978–2988, 2023.

[14] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[15] Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies, 2024. URL https://arxiv.org/abs/2402.04292.

[16] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[17] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[18] 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*, pages 991–1002. PMLR, 2022.

[19] Hanxiao Jiang, Binghao Huang, Ruihai Wu, Zhuoran Li, Shubham Garg, Hooshang Nayyeri, Shenlong Wang, and Yunzhu Li. Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation. *arXiv preprint arXiv:2402.15487*, 2024.

[20] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[21] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations, 2024. URL https://arxiv.org/abs/2402.10885.

[22] Weijie Kong, Zhaohui Lin, Wei Yu, Haotian Guo, Zhian Su, and Huixu Dong. Affpose: An integrated rgb-based framework for simultaneous pose estimation and affordance detection in robotic tool manipulation. *IEEE Robotics and Automation Letters*, 10(10):10170–10177, 2025. doi: 10.1109/LRA.2025.3598984.

[23] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies, 2016. URL https://arxiv.org/abs/1504.00702.

[24] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, 2016. URL https://arxiv.org/abs/1603.02199.

[25] Hang Li, Qian Feng, Zhi Zheng, Jianxiang Feng, Zhaopeng Chen, and Alois Knoll. Language-guided object-centric diffusion policy for generalizable and collision-aware manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12834–12841. IEEE, 2025.

[26] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025.

[27] Yu Li, Xiaojie Zhang, Ruihai Wu, Zilong Zhang, Yiran Geng, Hao Dong, and Zhaofeng He. Unidoormanip: Learning universal door manipulation policy over large-scale and diverse door manipulation environments. *arXiv preprint arXiv:2403.02604*, 2024.

[28] Zhaohui Lin, Haonan Dong, Weijie Kong, Haoran Huang, I-Ming Chen, and Huixu Dong. A coarse-to-fine multimodal detection framework based on deep learning for robotic coating tasks. *IEEE/ASME Transactions on Mechatronics*, 31(1):639–650, 2026. doi: 10.1109/TMECH.2025.3595263.

[29] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

[30] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[31] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation, 2019. URL https://arxiv.org/abs/1903.06684.

[32] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022. URL https://arxiv.org/abs/2112.03227.

[33] Yu Qi, Yuanchen Ju, Tianming Wei, Chi Chu, Lawson LS Wong, and Huazhe Xu. Two by two: Learning multi-task pairwise objects assembly for generalizable robot manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17383–17393, 2025.

[34] Jianing Qian, Yunshuang Li, Bernadette Bucher, and Dinesh Jayaraman. Task-oriented hierarchical object decomposition for visuomotor control. *arXiv preprint arXiv:2411.01284*, 2024.

[35] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone, 2022. URL https://arxiv.org/abs/2206.11251.

[36] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation, 2021. URL https://arxiv.org/abs/2109.12098.

[37] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[38] Andreas Sochopoulos, Nikolay Malkin, Nikolaos Tsagkas, João Moura, Michael Gienger, and Sethu Vijayakumar. Fast flow-based visuomotor policies via conditional optimal transport couplings, 2025. URL https://arxiv.org/abs/2505.01179.

[39] Zhian Su, Yicheng Ma, Haotian Guo, and Huixu Dong. Construction of bin-picking system for logistic application: A hybrid robotic gripper and vision-based grasp planning. *IEEE Robotics and Automation Letters*, 10(8): 8300–8307, 2025. doi: 10.1109/LRA.2025.3585393.

[40] Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. Kite: Keypoint-conditioned policies for semantic manipulation. *arXiv preprint arXiv:2306.16605*, 2023.

[41] 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.

[42] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[43] Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Rise: 3d perception makes real-world robot imitation simple and effective, 2024. URL https://arxiv.org/abs/2404.12281.

[44] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy, 2024. URL https://arxiv.org/abs/2407.01812.

[45] Shengjie Wang, Jiacheng You, Yihang Hu, Jiongye Li, and Yang Gao. Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation, 2025. URL https://arxiv.org/abs/2501.14400.

[46] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. URL https://arxiv.org/abs/2312.08344.

[47] Shangning Xia, Hongjie Fang, Cewu Lu, and Hao-Shu Fang. Cage: Causal attention enables data-efficient generalizable robotic manipulation, 2024. URL https://arxiv.org/abs/2410.14974.

[48] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning, 2025. URL https://arxiv.org/abs/2502.16932.

[49] Jingyun Yang, Zi ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim(3)-equivariant diffusion policy for generalizable and data efficient learning, 2024. URL https://arxiv.org/abs/2407.01479.

[50] Jingyun Yang, Congyue Deng, Jimmy Wu, Rika Antonova, Leonidas Guibas, and Jeannette Bohg. Equivact: Sim(3)-equivariant visuomotor policies beyond rigid object manipulation, 2024. URL https://arxiv.org/abs/2310.16050.

[51] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations, 2024. URL https://arxiv.org/abs/2403.03954.

[52] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation, 2022. URL https://arxiv.org/abs/2010.14406.

[53] Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation, 2024. URL https://arxiv.org/abs/2412.04987.

[54] Xinyu Zhang and Abdeslam Boularias. One-shot imitation learning with invariance matching for robotic manipulation, 2024. URL https://arxiv.org/abs/2405.13178.

[55] Haibo Zhao, Yu Qi, Boce Hu, Yizhe Zhu, Ziyan Chen, Heng Tian, Xupeng Zhu, Owen Howell, Haojie Huang, Robin Walters, Dian Wang, and Robert Platt. Generalizable hierarchical skill learning via object-centric representation, 2025. URL https://arxiv.org/abs/2510.21121.

[56] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[57] Huayi Zhou, Ruixiang Wang, Yunxin Tai, Yueci Deng, Guiliang Liu, and Kui Jia. You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations, 2025. URL https://arxiv.org/abs/2501.14208.

[58] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023.