EQUIKEY:OBJECT-CENTRIC SE(3)-EQUIVARIANT KEYPOSE GUIDANCE FOR ONE-SHOT VLA ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models acquire broad semantic and visuomotor priors from large-scale robot data, yet adapting them to novel manipulation tasks remains challenging when only a single demonstration is available. In particular, a generalist VLA may recognize task-relevant objects and reuse familiar motor skills while still failing to infer the precise three-dimensional configurations required for successful execution. We introduce EQUIKEY, an object-centric $\mathrm{SE}(3)$-equivariant keypose guidance framework for one-shot VLA adaptation. Given a single demonstration of a novel task, a lightweight task-specific $\mathrm{SE}(3)$-equivariant predictor learns to infer a sparse sequence of task-relevant keyposes. These predicted keyposes provide explicit geometric guidance to a pretrained keypose-conditioned EQUIKEY-VLA, whose backbone remains frozen during task adaptation. During execution, the predicted keypose sequence is temporally scheduled as high-level geometric targets, while EQUIKEY-VLA generates dense continuous actions conditioned on the latest observation. On 10 one-shot OOD tasks in RoboTwin 2.0, our method achieves average success rates of 77.1% and 76.9% under the Easy and Hard settings, improving the strongest VLA baseline by 42.5% and 44.6% and the strongest task-specific one-shot adaptation baseline by 26.5% and 33.6%, respectively. In real-world one-shot Task-OOD experiments, our method achieves a 73.3% success rate, substantially outperforming the strongest baseline at 11.7%. These results demonstrate that sparse $\mathrm{SE}(3)$-equivariant keypose guidance provides an effective interface for combining geometric inductive biases with the semantic and visuomotor capabilities of pretrained VLA models.

1 INTRODUCTION

Robotic foundation models have advanced rapidly in recent years, demonstrating promising performance in robotic manipulation (Kim et al., 2024; Black et al., 2025; Wang et al., 2026). Unlike traditional manipulation policies (Chi et al., 2025; Zhao et al., 2023; Su et al., 2025), Vision-Language-Action (VLA) models act as generalists pretrained on large-scale embodied datasets, thereby acquiring broad semantic and visuomotor priors (Black et al., 2025; Bjorck et al., 2025; Zheng et al., 2025; Physical Intelligence et al., 2026). However, existing VLA models exhibit limited generalization and lack zero-shot or few-shot transfer capabilities to novel tasks (Kim et al., 2025; Fei et al., 2025). A primary bottleneck to VLA generalization is the difficulty of identifying correct execution trajectories from the vast, high-dimensional, and previously unseen action distribution of a novel task (Li et al., 2025). Consequently, adapting to novel tasks heavily relies on task-specific fine-tuning with substantial demonstration datasets covering the target action distribution. Yet, unlike the internet-scale text or video corpora leveraged by LLMs, real-world robotic data remains remarkably scarce and expensive to acquire (Khazatsky et al., 2024). To address this challenge, we explore whether VLA models can achieve enhanced generalization and one-shot task adaptation without relying on scaling up backbone parameters or expanding pre-training datasets.

A promising paradigm for improving sample efficiency and generalization in robotic manipulation is encoding equivariance into the system (Huang et al., 2022; 2024). For a given manipulation task, the manipulated objects inherently possess spatial symmetries: when an object's $\mathrm{SE}(3)$ pose changes, the policy only needs to output the correspondingly transformed action, rather than inferring valid actions from a wide unseen robot state distribution. Therefore, introducing $\mathrm{SE}(3)$-equivariant modules into VLA enables the extraction of object-centric geometric priors from a one-shot demonstration

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.

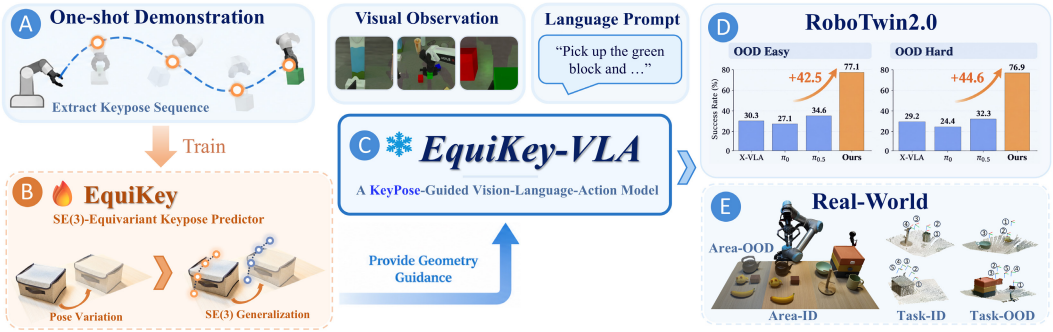


Figure 1: **Overview of EquiKey.** Given a single demonstration of a novel task, we automatically construct keypose supervision (A) to train EQUIKEY for predicting an SE(3)-equivariant keypose sequence (B). The predicted keyposes provide geometric guidance to the frozen keypose-conditioned EQUIKEY-VLA for action generation (C). Simulation results demonstrate strong one-shot adaptation capability (D), while real-world experiments further validate the effectiveness of our approach (E).

set, maintaining robust generalization under SE(3) spatial pose variations. By leveraging these SE(3)-equivariant geometric priors as guidance, pre-trained VLA models, which inherently possess strong task-level semantic understanding, can execute novel tasks without requiring costly fine-tuning of the large-scale backbones.

Based on these insights, we introduce **EQUIKEY**, an object-centric SE(3)-equivariant keypose guidance framework for one-shot VLA adaptation. To the best of our knowledge, EQUIKEY is the first framework to introduce SE(3)-equivariant keypose guidance into pretrained VLA models for one-shot task adaptation. Specifically, EQUIKEY predicts an SE(3)-equivariant keypose sequence as a geometric prior to guide the pretrained keypose-conditioned **EQUIKEY-VLA** during trajectory generation. We evaluate EQUIKEY-VLA on RoboTwin 2.0 simulation benchmark (Chen et al., 2025b) across 10 one-shot OOD tasks, where it achieves average success rates of 77.1% and 76.9% under Easy and Hard settings, outperforming the strongest VLA baseline by up to 44.6% and the strongest one-shot adaptation baseline by up to 33.6%. Real-world experiments further demonstrate average success rates of 66.7% on spatial OOD and 73.3% on one-shot Task-OOD. These results demonstrate that our method not only improves the spatial generalization capability of VLA models but also enables strong one-shot adaptation to novel tasks.

Our main contributions are summarized as follows:

1. We propose EQUIKEY, a lightweight SE(3)-equivariant keypose sequence predictor with fewer than 3M parameters that learns object-centric geometric priors from a single demonstration.
2. We present EQUIKEY-VLA, a novel architecture that conditions action generation on SE(3)-equivariant geometric guidance without requiring extensive fine-tuning.
3. We conduct evaluations in both simulation and real-world environments, demonstrating significant improvements in spatial generalization and one-shot adaptation over strong baselines.

2 RELATED WORK

Vision-Language-Action Models and Generalization. Vision-Language-Action (VLA) models aim to build generalist robotic policies by leveraging large-scale vision-language representations and embodied demonstration data for end-to-end action generation. Early autoregressive VLA models (Brohan et al., 2022; Zitkovich et al., 2023; Kim et al., 2024) discretize robot actions into tokens and generate them with LLM backbones, while recent VLA models synthesize continuous action chunks via flow matching (Black et al., 2025; 2024; Bjorck et al., 2025; Zheng et al., 2025; Shukor et al., 2025; Zhai et al., 2025). Despite their promising performance, existing models still require fine-tuning on tens or hundreds of demonstration trajectories for downstream tasks, lacking generalization to unseen tasks (Kim et al., 2025). Several recent studies have explored data-efficient adaptation using one-shot demonstrations (Li et al., 2025; Zhao et al., 2025; Chen et al., 2025a), but

these approaches primarily rely on skill-space calibration or video demonstration imitation, both of which require fine-tuning the VLA backbone. In contrast, our method does not directly fine-tune the VLA backbone to cover the action distribution of a novel task. Instead, it extracts task-relevant geometry priors from a single demonstration and uses an $\mathrm{SE}(3)$-equivariant predictor to guide action.

Equivariant Policy Learning in Robotics. Equivariance has been widely explored as an effective inductive bias for improving sample efficiency and spatial generalization in robotic manipulation (Seita et al., 2021; Huang et al., 2022), since many manipulation tasks exhibit strong spatial geometric structure. Prior studies have introduced $\mathrm{SE}(2)$ or $\mathrm{SE}(3)$ equivariant representations into imitation learning and reinforcement learning, showing strong performance (Wang et al., 2022a;b;c; Zhu et al., 2026; Gao et al., 2024; Yang et al., 2025; Wang et al., 2025; Tie et al., 2025). However, most existing equivariant policies are designed as task-specific low-level controllers or standalone policy architectures, and thus lack the open-vocabulary semantic understanding and broad visuomotor priors of pretrained VLA models. Recent work has introduced $\mathrm{SO}(2)$-equivariance into VLA models (Thien-Loc Ha, 2026), which only provides planar rotation equivariance. In contrast, our method introduces $\mathrm{SE}(3)$-equivariant keypose guidance into pretrained VLA models for one-shot task adaptation.

Keypose Guidance and Hierarchical Manipulation. The keypose formulation was first introduced by James & Davison (2022), which approximates continuous manipulator trajectories using a sequence of discrete keyposes, thereby simplifying policy learning. Inspired by this idea, many robotic policies do not directly predict continuous trajectories, but instead predict sparse keyposes as high-level guidance, while a low-level motion planner executes the corresponding manipulation trajectories (James et al., 2022; Shridhar et al., 2023; Gervet et al., 2023; Goyal et al., 2023; 2024; Xian et al., 2023). This hierarchical abstraction effectively shrinks the action search space and decouples long-horizon spatial planning from continuous motor control. However, such keypose-based policies struggle with tasks requiring reactive, high-frequency control, and they remain sample-inefficient when adapting to novel tasks. Inspired by this coarse-to-fine hierarchical paradigm, we leverage the strong geometric inductive biases of $\mathrm{SE}(3)$-equivariance to predict sparse keyposes as high-level guidance, enabling the generalist VLA model to generate fine-grained, high-frequency continuous action trajectories conditioned on this geometric prior.

3 METHOD

Our framework consists of two coupled components: a lightweight task-specific $\mathrm{SE}(3)$-equivariant keypose predictor, **EQUIKEY**, and a VLA policy capable of incorporating geometric prior guidance, **EQUIKEY-VLA**. Given a single task demonstration, we automatically construct keypose and object-centric supervision from the demonstration trajectory (Section 3.1). EQUIKEY learns to predict a sparse sequence of task-relevant geometric keyposes from the initial RGB-D observation (Section 3.2). During pretraining, EQUIKEY-VLA learns an embodiment-specific action representation while learning to exploit geometric priors through keypose conditioning (Section 3.3). For a new task, the frozen EQUIKEY-VLA uses the keyposes predicted by EQUIKEY as high-level geometric guidance, while generating dense continuous actions conditioned on the latest observation (Section 3.4).

3.1 AUTOMATIC CONSTRUCTION OF KEYPOSE SUPERVISION

Automatic Discovery of Keypose Sequences. To discover keyposes without manual effort, we employ a vision-language model (Qwen3.5-VL-27B) (Bai et al., 2025) to automatically identify critical kinematic milestones from demonstration videos. For each collected demonstration τ, the model processes the episode video V_τ and the corresponding instruction ℓ_τ to identify key moments associated with major kinematic transitions, such as establishing a grasp, initiating object transport, or releasing an object. Specifically, the model extracts a sequence of normalized relative timestamps, acting-arm indicators, and the corresponding stage-relevant manipulation objects, denoted as

$$\mathcal{A}_\tau = \{(r_i, b_i, n_i)\}_{i=1}^{M_\tau}, \tag{1}$$

where $r_i \in [0, 1]$ denotes the normalized timestamp, $b_i \in \{\texttt{left}, \texttt{right}, \texttt{dual}\}$ denotes the acting arm, and n_i denotes the manipulation object associated with the corresponding stage. Prompt constraints enforce a structured machine-readable format, requiring at least two key moments per

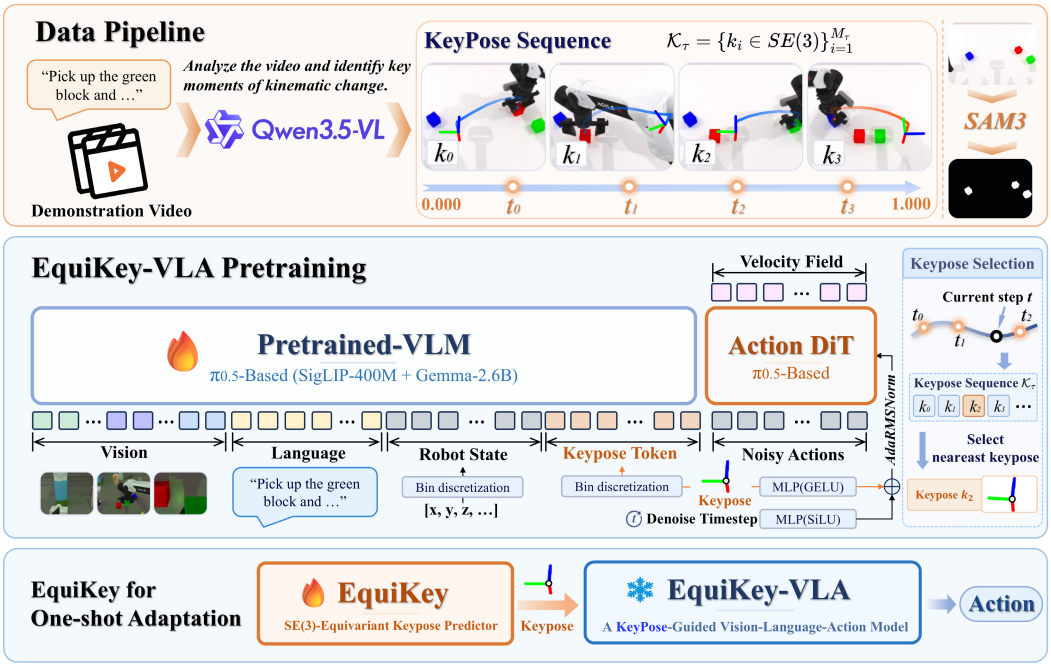


Figure 2: **Framework of the proposed method.** We construct keypose supervision, pretrain EQUIKEY-VLA, and adapt to a new task by training only the lightweight EQUIKEY predictor.

episode and guaranteeing the inclusion of the task completion state. Further details on the prompt design and structured annotation format are provided in Appendix A.

Each predicted timestamp is aligned with the synchronized robot end-effector trajectory to obtain the corresponding keypose. The resulting sequence

$$\mathcal{K}_\tau = \{k_i \in SE(3)\}_{i=1}^{M_\tau} \tag{2}$$

captures the task-relevant critical kinematic configurations demonstrated throughout the episode. The normalized timestamp r_i is retained for temporal scheduling during deployment (Section 3.4), while the acting-arm label b_i is retained as an auxiliary mask during policy conditioning (Section 3.3).

Object-Centric Visual Supervision. After obtaining the keypose sequence, we group consecutive keyposes associated with the same object name n_i into an object-bound stage. Specifically, let $\mathcal{I}_j$ denote the contiguous set of keypose indices in the j-th stage, such that $n_i = n^{(j)}$ for all $i \in \mathcal{I}_j$, where $n^{(j)}$ denotes the manipulation object associated with that stage. The corresponding keypose subsequence is

$$\mathcal{K}_\tau^{(j)} = \{k_i \in SE(3) \mid i \in \mathcal{I}_j\}. \tag{3}$$

To construct object-centric supervision for EQUIKEY, we use SAM3 (Carion et al., 2026) to automatically segment the corresponding object from the initial image of each training demonstration, with $n^{(j)}$ provided as the text prompt. We then lift the initial RGB-D observation into a colored point cloud $\mathcal{P} = \{(x_m, f_m)\}_{m=1}^{N}$ and project the soft segmentation mask onto the sampled 3D points, yielding a soft segmentation score $w_m^{(j)} \in [0, 1]$ for each point x_m. We normalize these scores into an object-centric soft distribution as

$$\tilde{w}_m^{(j)} = \frac{(w_m^{(j)} + \epsilon)^\kappa}{\sum_{m'=1}^{N} (w_{m'}^{(j)} + \epsilon)^\kappa}, \tag{4}$$

which serves as a training-only object-centric prior for the global score field of EQUIKEY.

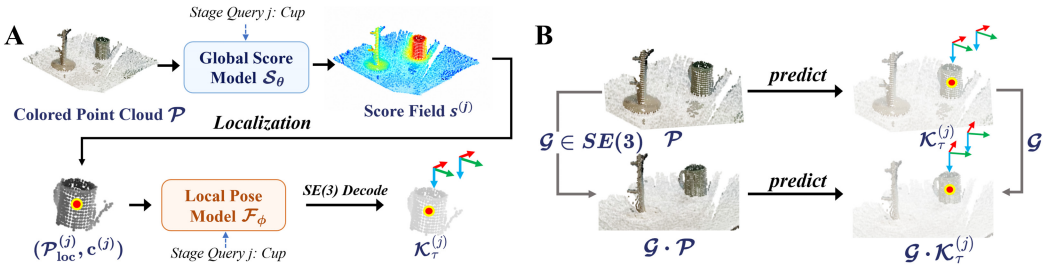


Figure 3: **Pipeline and equivariance of EQUIKEY.** EQUIKEY predicts keypose sequences from the initial RGB-D observation (A) and preserves SE(3)-equivariance under rigid transformations (B).

3.2 EQUIKEY: HIERARCHICAL SE(3)-EQUIVARIANT KEYPOSE PREDICTION

Given the colored point cloud $\mathcal{P}$ reconstructed from the initial RGB-D observation, EQUIKEY predicts one keypose subsequence for each object-bound stage. Inspired by the scalar-relevance and equivariant-pose decomposition of RiEMann (Gao et al., 2024), EQUIKEY separates stage-relevant localization from geometric pose prediction. Unlike RiEMann, its local branch performs hierarchical region reasoning and jointly predicts an ordered keypose subsequence. We denote the global score model by $\mathcal{S}_\theta$ and the local pose model by $\mathcal{F}_\phi$; their interface is

$$\mathbf{s}^{(j)} = \mathcal{S}_\theta(\mathcal{P}, j), \qquad (\mathcal{P}^{(j)}_{\text{loc}}, \mathbf{c}^{(j)}) = \mathcal{C}(\mathcal{P}, \mathbf{s}^{(j)}), \qquad \{(\hat{\delta}_i, u_i, v_i)\}_{i \in \mathcal{I}_j} = \mathcal{F}_\phi(\mathcal{P}^{(j)}_{\text{loc}}, j), \tag{5}$$

where $\mathcal{C}$ denotes deterministic score normalization, support selection, cropping, and centering.

Global score model. The SE(3)-Transformer score model (Fuchs et al., 2020) processes a voxelized radius graph of $\mathcal{P}$ and produces the stage-conditioned scalar field $\mathbf{s}^{(j)} = (s^{(j)}_1, \ldots, s^{(j)}_N)$. The scores are normalized over the selected support to obtain an object-centric center $\mathbf{c}^{(j)}$ and to crop the corresponding local geometry $\mathcal{P}^{(j)}_{\text{loc}}$. One score field is shared by all keypose slots in the same stage because it localizes their common manipulation object.

Local pose model. The local crop is centered at $\mathbf{c}^{(j)}$ and encoded using vector neurons (Deng et al., 2021). Point features are pooled into region tokens, and hierarchical relation attention models interactions between object parts. Its attention weights are computed from rotation-invariant scalars, while its values and relative-position messages remain equivariant vectors. The pooled descriptor is routed through stage-conditioned heads that jointly predict a translation offset $\hat{\delta}_i$ and two rotation vectors u_i, v_i for each $i \in \mathcal{I}_j$. They are decoded as

$$\hat{R}_i = \text{IMGS}(u_i, v_i), \qquad \hat{t}_i = \mathbf{c}^{(j)} + \hat{\delta}_i, \qquad \hat{k}_i = \begin{bmatrix} \hat{R}_i & \hat{t}_i \\ 0 & 1 \end{bmatrix} \in \text{SE}(3). \tag{6}$$

Here, IMGS denotes the Iterative Modified Gram–Schmidt orthonormalization (Zhou et al., 2019). The decoded poses form the stage-level subsequence $\hat{\mathcal{K}}^{(j)}_\tau = \{\hat{k}_i \mid i \in \mathcal{I}_j\}$.

Additional architectural details and training specifications of both the global score model and the local pose model are provided in Appendix B.

Training and equivariance. We first train $\mathcal{S}_\theta$ with the object-centric soft supervision constructed above, then freeze it and train $\mathcal{F}_\phi$ with keypose and relative-transform objectives. The supervision prior is absent at inference. Conditional on corresponding discretized geometry, scalar scores are preserved at corresponding transformed points and the resulting center transforms with the point cloud. The centered pose model is rotation equivariant and restores translation through $\mathbf{c}^{(j)}$. Thus a rigid transformation of the observed geometry induces the same transformation of the predicted keyposes. Formal statements, assumptions, and proofs of equivariance are given in Appendix C.

The stage-wise subsequences are concatenated in their annotated temporal order. During execution, the scheduling module selects one active keypose from this sequence at each timestep and supplies it to EQUIKEY-VLA as geometric guidance.

3.3 EQUIKEY-VLA: KEYPOSE-CONDITIONED VLA POLICY

Unified State-Action-Keypose Representation. To enable VLA models to better share representations across the state, action, and keypose spaces, we adopt a unified 20-dimensional representation for the robot state s_t, action a_t, and keypose condition. Specifically, for each arm, the representation consists of: (1) the absolute Cartesian end-effector (EEF) position $\mathbf{p} \in \mathbb{R}^3$, (2) the absolute EEF orientation $\mathbf{r} \in \mathbb{R}^6$ using the Rotate6D representation, and (3) the gripper state $g \in [0, 1]$, resulting in a 10-dimensional vector per arm.

EQUIKEY predicts the active-arm geometric keypose $\hat{k}_i \in SE(3)$. We denote its corresponding full policy-side representation at timestep t as $\tilde{k}_{t,i} \in \mathbb{R}^{20}$, constructed from the current robot state s_t by replacing only the active-arm EEF pose with the EQUIKEY prediction, while retaining the remaining state components, including the inactive-arm pose and gripper states, from the current observation. For notational simplicity, we use $k_t^\star$ to denote the resulting 20-dimensional active keypose condition in the following VLA conditioning and execution equations.

Keypose Token Conditioning. We choose $\pi_{0.5}$ (Black et al., 2025) as the base VLA model, which is built on a PaliGemma-based (Beyer et al., 2024) mixture-of-transformers (MoT) architecture consisting of an understanding expert and an action expert. The understanding expert consists of a PaliGemma-2B backbone that processes the current images, task instruction, and robot state to extract semantic and contextual information. The action expert is implemented using a PaliGemma-300M backbone to generate action chunks through flow matching. To incorporate object-level geometric guidance, we investigate injecting keyposes into the understanding expert, the action expert, and both via dual-stream conditioning.

For injection into the understanding expert, we extend the original prefix textual prompt from "Task: $\{\ell_\tau\}$, State: $\{s_t\}$; Action:" to "Task: $\{\ell_\tau\}$, State: $\{s_t\}$, Keypose: $\{k_t^\star\}$; Action:", where $k_t^\star$ denotes the active keypose selected at timestep t by the scheduling mechanism described in Section 3.4. To align with the robot-state encoding scheme of $\pi_{0.5}$, each keypose is normalized using the corresponding 1% and 99% quantile statistics and then discretized into 256 uniformly spaced bins before being serialized into the prompt.

For the action expert, inspired by conditioning techniques used in controllable diffusion transformers (Dhariwal & Nichol, 2021; Peebles & Xie, 2023), we project the normalized active keypose through a masked MLP and add it to the denoising time embedding, which further modulates the AdaRMSNorm layers in the Transformer blocks of the action expert. Specifically, let $\bar{k}_t^\star$ denote the normalized active keypose, $m_t^k \in \{0, 1\}$ its validity mask, and e_λ the denoising time embedding at flow time λ. The AdaRMSNorm condition is computed as

$$\tilde{e}_{\lambda,t} = e_\lambda + m_t^k E_k(\bar{k}_t^\star), \tag{7}$$

$$\mathrm{AdaRMSNorm}_d(h_d; \tilde{e}_{\lambda,t}) = \mathrm{RMSNorm}(h_d) \odot (\mathbf{1} + \gamma_d(\tilde{e}_{\lambda,t})) + \beta_d(\tilde{e}_{\lambda,t}), \tag{8}$$

where E_k is an MLP that projects the keypose into the hidden dimension of the action expert, and γ_d and β_d are learned modulation functions for the d-th Transformer block. When no valid keypose is available, $m_t^k = 0$, and $\tilde{e}_{\lambda,t} = e_\lambda$, reducing the conditioning to the original time embedding.

Based on ablation findings, we select conditioning solely on the understanding expert as our primary model configuration. Further architectural details are provided in Appendix D.

Robust Conditioning via Modality Dropout. During training, we apply random condition dropout to prevent the policy from over-relying on any single conditioning signal. Specifically, with probability $p_k = 0.1$, the keypose condition is removed by omitting the Keypose field from the textual prompt. Independently, with probability $p_{vl} = 0.1$, the visual-language condition is dropped by replacing the input images with zero-valued images and the task instruction with an empty string. This training strategy encourages the policy to remain robust when geometric or visual-language cues are partially unavailable or unreliable.

3.4 TIME-SCHEDULED SE(3)-EQUIVARIANT KEYPOSE GUIDANCE FOR VLA EXECUTION

During task execution, we combine the SE(3)-equivariant keypose predictor with the keypose-conditioned VLA policy through a time-based scheduling mechanism. Given the initial scene

Table 1: **Main results on RoboTwin 2.0 simulation.** We report the average success rate (%) across 10 OOD tasks. Detailed per-task results are provided in Appendix 5 and Table 6.

Method	Data Protocol	VLA Updated?	Adapt. Params.	OOD Easy	OOD Hard
π_0	Mixed	–	–	27.1	24.4
$\pi_{0.5}$	Mixed	–	–	34.6	32.3
X-VLA	Mixed	–	–	30.3	29.2
$\pi_{0.5}$ + FLA	1 demo	Yes	4.7M	36.6	41.8
$\pi_{0.5}$ + FTM	1 demo	Yes	4K	43.2	40.7
$\pi_{0.5}$ + LoRA	1 demo	Yes	467M	<u>50.6</u>	<u>43.3</u>
Ours	1 demo	No	3M	**77.1**	**76.9**

observation o_0, EQUIKEY is executed once to predict the ordered task-level keypose sequence $\hat{\mathcal{K}}_\tau = \{\hat{k}_i \in SE(3)\}_{i=1}^{M_\tau}$, where each predicted keypose $\hat{k}_i$ is associated with the normalized timestamp $r_i \in [0, 1]$ obtained from the demonstration annotation process.

During execution, the normalized timestamps are mapped to the expected task horizon T_τ, estimated as the average trajectory length observed during data collection for task τ. At timestep t, we define the normalized execution progress and select the next scheduled keypose index as

$$\rho_t = \min\left(\frac{t}{T_\tau}, 1\right), \qquad \sigma(t) = \min\{i \mid \rho_t \le r_i\}. \tag{9}$$

The corresponding active keypose condition is $k_t^\star = \tilde{k}_{t,\sigma(t)}$, with the final keypose retained once the execution progress reaches the last scheduled timestamp.

At timestep t, the VLA acts as a low-level policy conditioned on the current observation, task instruction, and active keypose:

$$a_t \sim \pi_\theta(\cdot \mid o_t, \ell_\tau, k_t^\star). \tag{10}$$

EQUIKEY therefore provides a temporally scheduled sequence of geometric targets, while the VLA translates each active target into continuous actions using the latest observation.

4 EXPERIMENTS

In this section, we conduct both simulation and real-world experiments to evaluate the effectiveness of our proposed EQUIKEY and EQUIKEY-VLA. Our experiments are designed to address three questions: (1) Can EQUIKEY-VLA enable effective one-shot adaptation to unseen tasks compared with strong baselines? (2) Does $\mathrm{SE}(3)$-equivariant keypose guidance improve spatial generalization under distribution shifts? (3) How do keypose prediction quality and different conditioning strategies affect downstream policy performance?

4.1 SIMULATION EXPERIMENTS

Setup. We evaluate the proposed method on the RoboTwin 2.0 benchmark (Chen et al., 2025b), which contains 50 diverse bimanual manipulation tasks with substantial variations in object poses and scene configurations. For each task, the benchmark provides 50 demonstrations collected in clean scenes and 500 demonstrations collected under extensive scene randomization. To evaluate one-shot task adaptation, we randomly select 40 tasks for shared VLA training as the in-distribution (ID) split and reserve the remaining 10 tasks as the out-of-distribution (OOD) split (see Appendix E). For each OOD task, only one demonstration is provided for task-specific adaptation. EQUIKEY-VLA trains a lightweight SE(3)-equivariant keypose predictor from this demonstration while keeping the shared VLA policy frozen. We report the average Success Rate (SR) over 100 evaluation rollouts per task under both Easy and Hard settings.

Baselines. We compare against π_0, $\pi_{0.5}$, and X-VLA (Black et al., 2024; 2025; Zheng et al., 2025), trained on the mixed dataset containing the 40 ID tasks and one demonstration from each OOD

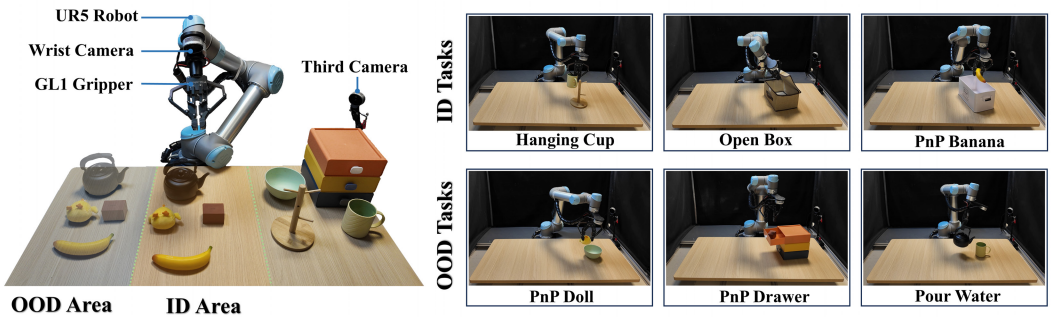


Figure 4: **Real-world experimental setup.** The left panel shows the hardware platform and the ID/OOD areas. The right panel shows the three ID tasks and three OOD tasks. All demonstrations are collected in the ID area, while evaluation is conducted in both the ID/OOD areas.

task. We further include task-specific one-shot adaptation baselines based on $\pi_{0.5}$: LoRA, FLA, and FTM (Li et al., 2026b). All one-shot methods use exactly the same single demonstration.

Results. As shown in Table 1, existing VLA baselines exhibit limited generalization to OOD tasks, with the strongest baseline, $\pi_{0.5}$, achieving only 34.6%/32.3% on OOD Easy/Hard. Task-specific one-shot adaptation improves performance, with $\pi_{0.5}$ +LoRA reaching 50.6%/43.3%. In comparison, EQUIKEY-VLA achieves 77.1%/76.9%, outperforming the strongest one-shot baseline by 26.5%/33.6% while keeping the VLA frozen. These results demonstrate the effectiveness of SE(3)-equivariant keypose guidance for one-shot adaptation to novel tasks.

4.2 REAL-WORLD EXPERIMENTS

Setup. We further evaluate EQUIKEY-VLA on a real-world robotic platform consisting of a UR5 arm equipped with a parallel-jaw gripper and two RealSense L515 cameras. We consider six manipulation tasks: *1) Hanging Cup*; *2) Open Box*; *3) PnP Banana*; *4) PnP Doll*; *5) PnP Drawer*; *6) Pour Water*. Tasks 1–3 are used as the task in-distribution (**Task-ID**) split, with 50 demonstrations collected for each task. Tasks 4–6 are treated as the task out-of-distribution (**Task-OOD**) split, where only one demonstration is collected for each task to evaluate one-shot adaptation.

Orthogonally, each task is evaluated under two spatial settings. In the **Area-ID** setting, objects are initialized within the spatial distribution covered by the demonstrations, whereas **Area-OOD** evaluates held-out workspace regions outside the demonstrated spatial distribution. VLA baselines are trained on the pooled dataset across all tasks, whereas our method trains the $\mathrm{SE}(3)$-equivariant keypose predictor using only the one-shot OOD demonstration without fine-tuning the VLA backbone. More details of the real-world setup are provided in Appendix F.

Results. Table 2 reports real-world success rates under task-level and spatial distribution shifts. The VLA baselines achieve high success rates only in the fully in-distribution setting, where the strongest baseline $\pi_{0.5}$ reaches 93.3%. Once objects move outside the demonstrated spatial distribution, performance drops sharply: on the same Task-ID tasks, the best baseline falls to 10.0% in Area-OOD, while X-VLA fails entirely. One-shot adaptation to Task-OOD is limited for the baselines, with the best performance reaching only 11.7%.

Table 2: **Real-world evaluation results.** We report average success rates (%) under task-level and spatial distribution shifts. For Task-ID, Area-ID and Area-OOD are each evaluated over 10 rollouts. For Task-OOD, each task is evaluated over 20 rollouts.

Setting	π_0	$\pi_{0.5}$	X-VLA	Ours
Task-ID	48.3	<u>51.7</u>	36.7	**78.3**
+ Area-ID	<u>90.0</u>	**93.3**	73.3	<u>90.0</u>
+ Area-OOD	6.7	<u>10.0</u>	0.0	**66.7**
Task-OOD	10.0	<u>11.7</u>	3.3	**73.3**

In contrast, EQUIKEY-VLA achieves 90.0% on Area-ID and retains 66.7% success on Area-OOD for the Task-ID tasks, demonstrating substantially stronger spatial generalization. For one-shot Task-OOD adaptation, the strongest baseline achieves only 11.7%, whereas EQUIKEY-VLA reaches 73.3%, an improvement of 61.6% while keeping the VLA backbone frozen. Overall, EQUIKEY-VLA

achieves 78.3% on Task-ID and 73.3% on Task-OOD, demonstrating that $\mathrm{SE}(3)$-equivariant keypose guidance improves spatial generalization while enabling data-efficient adaptation to novel tasks.

4.3 Ablation Study and Robustness Analysis

Predicted vs. Oracle Keypose Guidance. To analyze the performance ceiling imposed by keypose guidance, we compare EquiKey-VLA using our predicted keyposes against oracle variant. The oracle variant utilizes ground-truth keyposes extracted directly from demonstrations, providing an upper bound for the proposed guidance mechanism. As shown in Table 3, using predicted keyposes results in slight performance drops of only 8.6% and 7.2% on the OOD Easy and OOD Hard settings, respectively. This narrow gap confirms that our SE(3)-equivariant keypose predictor delivers sufficiently accurate geometric priors to guide VLA trajectory generation.

Table 3: **Ablation study.** Average success rate (%) over the 10 OOD tasks of RoboTwin 2.0.

Variant	OOD Easy	OOD Hard
Oracle	85.7	84.1
Ours	**77.1**	**76.9**
Dual Stream	73.8	75.4
Action Expert	59.7	61.4
w/o Keypose	34.6	32.3
w/o Dropout	72.7	71.1
Reach Scheduler	61.3	59.0
Interp. Planning	55.6	53.5

Ablation on Keypose Conditioning Strategies. We investigate how different keypose conditioning strategies affect one-shot adaptation on the OOD tasks. Removing keypose guidance entirely reduces the model to the standard $\pi_{0.5}$ baseline, resulting in a performance drop from 77.1% to 34.6% (-42.5%). We further compare the proposed understanding-expert conditioning with alternative dual-stream and action-expert conditioning strategies. The results show that injecting the keypose through the understanding expert provides the most effective interface for incorporating geometric guidance into the pretrained VLA. Finally, removing random modality dropout causes a performance drop from 77.1% to 72.7% (-4.4%), as it impairs the policy's resilience when visual observations or geometric cues are partially degraded.

Ablation on Scheduling and Control. We further investigate the roles of keypose scheduling and the VLA policy. Replacing our time-based scheduler with a reach-based scheduler reduces performance from 77.1%/76.9% to 61.3%/59.0% on OOD Easy/Hard. Directly interpolating between predicted keyposes without the VLA further yields 55.6%/53.5%, suggesting that sparse geometric targets alone are insufficient for robust manipulation. The VLA enables adjustment of the intermediate execution based on current observations, rather than mechanically tracking an interpolated trajectory between keyposes. To complement these downstream ablations, we also evaluate the geometric accuracy of the predicted keyposes. More details are provided in Appendix E.1.

5 Conclusion

In this paper, we introduced EquiKey, a novel framework that incorporates $\mathrm{SE}(3)$-equivariant geometric guidance into pre-trained Vision-Language-Action (VLA) models for one-shot task adaptation. By leveraging a lightweight, task-specific $\mathrm{SE}(3)$-equivariant predictor, EquiKey learns object-centric geometric priors from a single demonstration and uses the predicted keyposes to guide a frozen EquiKey-VLA. Extensive evaluations on the RoboTwin 2.0 and real-world robotic tasks demonstrate strong generalization to out-of-distribution (OOD) settings, improving the strongest VLA baseline by up to 44.6% in simulation and achieving 73.3% success on real-world one-shot Task-OOD tasks. These results highlight the potential of combining $\mathrm{SE}(3)$-equivariant geometric inductive biases with pretrained VLA models for generalizable and sample-efficient robotic manipulation.

Limitations. Despite its effectiveness, EquiKey has several limitations. EquiKey predicts the keypose sequence only once from the initial observation, resulting in semi-open-loop high-level guidance that cannot adapt to substantial scene changes during execution. Sparse keypose guidance may also be insufficient for highly dynamic, deformable, or contact-rich tasks. Moreover, the underlying VLA is not strictly $\mathrm{SE}(3)$-equivariant, and our evaluations remain limited in scale. Future work could explore closed-loop keypose re-prediction and end-to-end equivariant VLA architectures.

AI USE STATEMENT

Generative AI tools and coding agents were utilized for manuscript polishing, including proofreading, grammar correction, and stylistic refinements, as well as for assisting with code implementation and refactoring. All core research ideas, mathematical formulations, algorithmic designs, experimental methodologies, and analytical conclusions were independently conceived by the authors. All AI-assisted text and code were carefully reviewed and verified by the authors, who take full responsibility for the content of this work.

ETHICS STATEMENT

This research utilizes publicly available benchmark datasets released under appropriate open-source licenses, alongside additional data collected by the authors. No human subject studies were involved, and no personally identifiable information, sensitive, or harmful data was collected. All physical robotic experiments were conducted in compliance with standard institutional safety protocols for academic research.

REPRODUCIBILITY STATEMENT

To comply with the double-blind review policy, we plan to release our code and model weights after the review process. Detailed model architectures are provided in Appendix B and D. The simulation setup and RoboTwin 2.0 evaluation details are provided in Appendix E, while the real-world experimental setup and task details are provided in Appendix F.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π0: A vision-language-action flow model for general robot control. *arXiv preprint ARXIV.2410.24164*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

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.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Guangyan Chen, Meiling Wang, Qi Shao, Zichen Zhou, Weixin Mao, Te Cui, Minzhao Zhu, Yinan Deng, Luojie Yang, Zhanqi Zhang, et al. See once, then act: Vision-language-action model with task learning from one-shot video demonstrations. *arXiv preprint arXiv:2512.07582*, 2025a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025b.

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.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12200–12209, 2021.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33: 1970–1981, 2020.

Chongkai Gao, Zhengrong Xue, Shuying Deng, Tianhai Liang, Siqi Yang, Lin Shao, and Huazhe Xu. Riemann: Near real-time se (3)-equivariant robot manipulation without point cloud segmentation. *arXiv preprint arXiv:2403.19460*, 2024.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

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*, pp. 694–710. PMLR, 2023.

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.

Haojie Huang, Dian Wang, Robin Walters, and Robert Platt. Equivariant transporter network. *arXiv preprint arXiv:2202.09400*, 2022.

Haojie Huang, Owen Lewis Howell, Dian Wang, Xupeng Zhu, Robert Platt, and Robin Walters. Fourier transporter: Bi-equivariant robotic manipulation in 3d. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=UulwvAU1W0.

Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13729–13738. IEEE, 2022.

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

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

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026a.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, et al. Controlvla: Few-shot object-centric adaptation for pre-trained vision-language-action models. *arXiv preprint arXiv:2506.16211*, 2025.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. Vla models are more generalizable than you think: Revisiting physical and spatial modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35025–35035, 2026b.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4568–4575. IEEE, 2021.

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

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

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.

Trong-Bao Ho Long Dinh Minh Duc Nguyen Gia-Binh Nguyen Pham Tri Quang Minh N. Vu Duy M. H. Nguyen An Thai Le Ngo Anh Vien Thien-Loc Ha, Quang-Tan Nguyen. Equivla: A general framework for rotationally equivariant vision-language-action models. *arXiv preprint arXiv:2606.19784*, 2026.

Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. Et-seed: Efficient trajectory-level se(3) equivariant diffusion policy. In *International Conference on Learning Representations*, 2025.

Dian Wang, Mingxi Jia, Xupeng Zhu, Robin Walters, and Robert Platt. On-robot learning with equivariant models. *arXiv preprint arXiv:2203.04923*, 2022a.

Dian Wang, Robin Walters, and Robert Platt. $\mathrm{SO}(2)$-equivariant reinforcement learning. *arXiv preprint arXiv:2203.04439*, 2022b.

Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant q learning in spatial action spaces. In *Conference on Robot Learning*, pp. 1713–1723. PMLR, 2022c.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. In *Conference on Robot Learning*, 2025.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026.

Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023.

Jingyun Yang, Ziang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim(3)-equivariant diffusion policy for generalizable and data efficient learning. In *Conference on Robot Learning*, 2025.

Andy Zhai, Brae Liu, Bruno Fang, Chalse Cai, Ellie Ma, Ethan Yin, Hao Wang, Hugo Zhou, James Wang, Lights Shi, et al. Igniting vlms toward the embodied space. *arXiv preprint arXiv:2509.11766*, 2025.

Ruihan Zhao, Tyler Ingebrand, Sandeep Chinchali, and Ufuk Topcu. Mos-vla: A vision-language-action model with one-shot skill adaptation. *arXiv preprint arXiv:2510.16617*, 2025.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5738–5746. IEEE, 2019.

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

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A KEYPOSE ANNOTATION

Prompt for Keypose Annotation

System Content:
You are an expert video analyst for robot manipulation tasks. Your task is to analyze a sequence of frames and identify the critical moments of kinematic change with the manipulation object associated with each moment.

User Content:
Task description: `{task_description}`. This is a robotic manipulation task. Analyze the video and extract the critical moments. You MUST autonomously identify the critical moments of kinematic change, such as the gripper closing to grasp an object, or placing an object onto a surface. Rules:

1. Timestamps MUST be unquoted floats (e.g., 0.748) with exactly three decimal places, representing the relative time from 0.000 to 1.000.
2. You MUST generate at least two keyframes (the 'keyframes' list must contain 2 or more items).
3. For 'arm', identify the acting arm. You MUST strictly use one of these string values: "left", "right", or "dual".
4. For 'object', specify the manipulation object associated with the corresponding key moment using a concise object name.

Respond ONLY with a valid JSON strictly matching this format:

```
"key_moments": [
  {
    "timestamp": <FLOAT>,
    "arm": "<left_or_right_or_dual>",
    "object": "<manipulation_object>"
  },
  {
    "timestamp": <FLOAT>,
    "arm": "<left_or_right_or_dual>",
    "object": "<manipulation_object>"
  },
  ...
]
```

To automatically obtain critical kinematic keyposes without manual effort, we leverage Qwen3.5-VL-27B (Bai et al., 2025) as an offline video annotator. The VLM takes the episode video frames and the corresponding task instruction as input, and outputs a structured JSON dictionary containing the acting-arm designations, manipulation object names, and normalized relative completion timestamps (ranging from 0.000 to 1.000) for key kinematic milestones. Each predicted timestamp is then aligned with the synchronized robot end-effector trajectory to extract the corresponding robot keypose. The complete system prompt is presented in the box above.

B EQUIKEY ARCHITECTURE AND IMPLEMENTATION DETAILS

We decompose the model into a global score field for semantic localization and a local hierarchical pose network for geometric keypose prediction. The full architecture is summarized in Fig. 5: panel A shows global score-field localization, while panel B shows local pose prediction and its two expanded geometric reasoning modules.

B.1 SE(3)-EQUIVARIANT GLOBAL SCORE FIELD

Input graph. Given a global point cloud $\mathcal{P} = \{(x_m, f_m)\}_{m=1}^{N}$, the spatial coordinates $x_m \in \mathbb{R}^3$ define the graph geometry, while $f_m \in \mathcal{V}_0^{\oplus 3}$ contains RGB values. RGB channels are treated as type-0 scalars because a spatial rotation changes point positions but not their colors. We first voxelize the input at a resolution of $1\,\mathrm{cm}$ and construct a radius graph with threshold $6\,\mathrm{cm}$. Each edge (m, n) carries the relative position $r_{mn} = x_n - x_m$ and its invariant norm $\|r_{mn}\|_2$. Spherical-harmonic bases computed from r_{mn} couple input and output irreps in the SE(3)-Transformer convolutions. Because

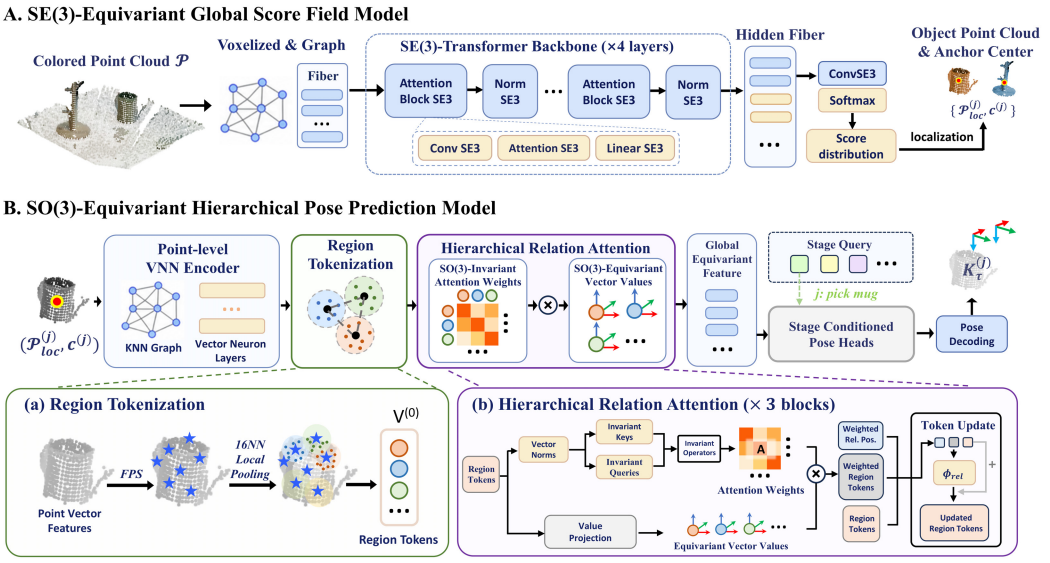


Figure 5: Overview of the EquiKey architecture. (A) The global SE(3)-equivariant score field localizes the task-relevant object and determines the crop center. (B) The local SO(3)-equivariant pose network encodes the object point cloud, constructs region tokens, performs hierarchical relation attention, and predicts stage-conditioned keyposes. Insets (a) and (b) detail region tokenization and invariant-gated equivariant relation attention.

all geometric messages depend on relative positions, the graph network is translation invariant and rotation equivariant for a fixed corresponding graph.

SE(3)-Transformer backbone. The backbone contains four single-head SE(3)-attention blocks, each followed by an equivariant normalization layer. Its hidden fiber is

$$\mathcal{H} = \bigoplus_{\ell=0}^{3} \left(\mathbb{R}^8 \otimes \mathcal{V}_\ell\right).$$

A final equivariant convolution projects the hidden fiber to type-0 output channels, one for each object-bound stage. For stage j, we write the selected scalar field as

$$\mathbf{s}^{(j)} = \mathcal{S}_\theta(\mathcal{P}, j) = \{s_m^{(j)}\}_{m=1}^{N_v}, \qquad s_m^{(j)} \in \mathcal{V}_0,$$

where N_v is the number of voxelized points. The stage query selects an output channel but carries no spatial representation. Since the output is type-0, rotating the input geometry preserves the score attached to each corresponding point.

Score readout and localization. Given the stage-conditioned scalar field $\mathbf{s}^{(j)}$ produced by the SE(3)-Transformer, we apply a masked softmax over valid nodes to obtain normalized point weights. At deployment, we deterministically retain the high-score support and renormalize the corresponding weights as $\widetilde{\alpha}_m^{(j)}$. The object-centric localization center is then computed as $\mathbf{c}^{(j)} = \sum_m \widetilde{\alpha}_m^{(j)} x_m$. We collect score normalization, support selection, high-resolution cropping, and centering into the deterministic operator

$$(\mathcal{P}_{\text{loc}}^{(j)}, \mathbf{c}^{(j)}) = \mathcal{C}(\mathcal{P}, \mathbf{s}^{(j)}). \tag{11}$$

B.2 Hierarchical Equivariant Pose Prediction

Given $\mathcal{P}_{\text{loc}}^{(j)}$ and stage j, the local pose model $\mathcal{F}_\phi$ predicts the corresponding ordered keypose subsequence. It uses only the selected geometry. As shown in panel B of Fig. 5, $\mathcal{F}_\phi$ contains a point-level vector-neuron (VN) encoder, a hierarchical region abstraction, an equivariant relation encoder, and stage-conditioned pose heads.

Equivariant feature representation. The local encoder represents each feature channel as a type-1 vector $\mathbf{v}_c \in \mathcal{V}_1 \simeq \mathbb{R}^3$ and employs Vector Neuron (VN) layers (Deng et al., 2021) throughout the network. We denote the space of C type-1 vector channels by $\mathcal{W}_C = \mathbb{R}^C \otimes \mathcal{V}_1$. A C-channel feature is written as $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_C] \in \mathbb{R}^{C\times 3}$, with rotations acting jointly on the vector dimension. VN operations preserve $\mathrm{SO}(3)$ equivariance, while norms and inner products provide rotation-invariant scalar quantities used by the subsequent relation modules.

Localized and centered object geometry. For stage j, the localization operator $\mathcal{C}$ extracts the local point cloud $\mathcal{P}_{\mathrm{loc}}^{(j)}$ and its geometric reference center $\mathbf{c}^{(j)}$. The crop is padded or subsampled to $N_{\mathrm{loc}} = 512$ points and represented in centered coordinates $\bar{x}_m = (x_m - \mathbf{c}^{(j)})/r$, with $r = 0.12\,\mathrm{m}$. A binary validity mask $\mu_m \in \{0, 1\}$ excludes padded points from graph construction and pooling. Localization quality affects which geometry is observed, but the pose model is conditional only on the resulting crop and stage index.

Point-level VN encoding and region tokenization. We construct a Euclidean k-nearest-neighbor graph with $k = 8$ on the valid centered points. For every directed edge (m, n), the input feature is

$$\mathbf{e}_{mn} = \left[\widehat{\bar{x}}_m \times \bar{x}_n, \bar{x}_n - \bar{x}_m, \bar{x}_m\right] \in \mathcal{V}_1^{\oplus 3},$$

where $\widehat{\bar{x}}_m = \bar{x}_m/\|\bar{x}_m\|_2$ denotes the normalized local direction.

A VN layer maps this feature to 128 vector channels and averages messages over the neighbors. Five subsequent VN blocks combine point-wise features with a masked global vector summary. Their outputs are concatenated and projected to $\mathbf{H} \in \mathbb{R}^{N_{\mathrm{loc}}\times C\times 3}$ with $C = 128$.

To reduce point-level redundancy, we select $G = 32$ region centers $\{\mathbf{c}_g\}_{g=1}^G$ by masked farthest-point sampling. Each region pools the features of its 16 nearest valid points $\mathcal{N}_g$, as expanded in panel B(a) of Fig. 5:

$$\mathbf{v}_g^{(0)} = \phi_{\mathrm{reg}}\left(\left[\frac{1}{|\mathcal{N}_g|}\sum_{m\in\mathcal{N}_g}\mathbf{H}_m, \mathbf{c}_g\right]\right), \tag{12}$$

where ϕ_{reg} is a $\mathrm{VN}_{C+1\to C}$ layer. The resulting region tokens $\mathsf{V}^{(0)} = (\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_G^{(0)})$ form a feature tensor in $\mathbb{R}^{G\times C\times 3}$.

Hierarchical relation attention. We apply three relation blocks to the region tokens. Each block uses invariant scalars to determine which regions should communicate, while the transmitted values remain type-1 vectors. Panel B(b) of Fig. 5 visualizes this invariant-gated equivariant message passing. For region g, let $\mathbf{v}_{g,c}$ denote its c-th vector channel. Vector norms define the invariant descriptor

$$\mathbf{u}_g = [\|\mathbf{v}_{g,1}\|_2, \ldots, \|\mathbf{v}_{g,C}\|_2] \in \mathbb{R}^C.$$

For attention head h, shared affine maps produce scalar queries and keys,

$$\mathbf{q}_{hg} = \mathbf{W}_h^Q\mathbf{u}_g + \mathbf{b}_h^Q, \qquad \mathbf{k}_{hg} = \mathbf{W}_h^K\mathbf{u}_g + \mathbf{b}_h^K.$$

The inter-region attention logit is

$$\ell_{hgg'} = \frac{\mathbf{q}_{hg}^\top\mathbf{k}_{hg'}}{\sqrt{d_h}} + b_h(\|\mathbf{c}_{g'} - \mathbf{c}_g\|_2^2) + \frac{\gamma_h}{C}\sum_{c=1}^{C}\mathbf{v}_{g,c}^\top\mathbf{v}_{g',c},$$

where $d_h = 32$, $b_h(\cdot)$ is a learned distance-bias function, and γ_h is a learned coefficient. Every term is rotation invariant. Hence $A_{hgg'} = \mathrm{softmax}_{g'}(\ell_{hgg'})$ is a type-0 scalar.

We average the four attention heads to obtain $A_{gg'} = \frac{1}{4}\sum_h A_{hgg'}$, forming $\mathsf{A} \in \mathbb{R}^{G\times G}$. A type-1 value projection $\psi_V : \mathcal{W}_C \to \mathcal{W}_C$ is applied independently to each region, after which A acts only on the region axis:

$$\overline{\mathbf{v}}_g = \sum_{g'=1}^{G} A_{gg'}\psi_V(\mathbf{v}_{g'}), \qquad \overline{\mathbf{r}}_g = \sum_{g'=1}^{G} A_{gg'}(\mathbf{c}_{g'} - \mathbf{c}_g).$$

The region token is updated by

$$\mathbf{v}'_g = \mathbf{v}_g + \phi_{\mathrm{rel}}([\mathbf{v}_g, \overline{\mathbf{v}}_g, \overline{\mathbf{r}}_g]), \tag{13}$$

where ϕ_{rel} is a $\mathrm{VN}_{2C+1\to C}$ layer. Invariant attention weights therefore gate equivariant vector messages without changing their representation type.

Global equivariant feature and pose decoding. For each final token, let $\mathbf{u}_g^{(3)} = [\|\mathbf{v}_{g,1}^{(3)}\|_2, \dots, \|\mathbf{v}_{g,C}^{(3)}\|_2]$. An invariant MLP scores these norm descriptors, and the resulting softmax weights are used to pool the region features into a global equivariant descriptor $\mathbf{z} \in \mathcal{W}_C$. We additionally augment this descriptor with equivariant second- and third-order geometric moments,

$$\mathbf{\Sigma} = \frac{1}{N_{\text{valid}}} \sum_m \mu_m \bar{x}_m \bar{x}_m^\top, \qquad \boldsymbol{\mu}_3 = \frac{1}{N_{\text{valid}}} \sum_m \mu_m \|\bar{x}_m\|_2^2 \bar{x}_m, \tag{14}$$

where $N_{\text{valid}} = \sum_m \mu_m$. These geometric features are fused as

$$\widetilde{\mathbf{z}} = \phi_{\text{mom}}([\mathbf{z}, \mathbf{\Sigma}\mathbf{z}, \boldsymbol{\mu}_3]) \in \mathcal{W}_C, \tag{15}$$

where ϕ_{mom} denotes an equivariant VN fusion layer.

The resulting descriptor is routed to stage-specific pose heads that predict a translation offset $\widehat{\boldsymbol{\delta}}_i$ and two type-1 rotation vectors $\mathbf{u}_i, \mathbf{v}_i$ for each $i \in \mathcal{I}_j$. The translation offsets are scaled by 0.45 and added to the geometric reference center to obtain $\widehat{\mathbf{t}}_i = \mathbf{c}^{(j)} + \widehat{\boldsymbol{\delta}}_i$. The rotation vectors are converted to a right-handed rotation $\widehat{\mathbf{R}}_i = \text{IMGS}(\mathbf{u}_i, \mathbf{v}_i)$ using Iterative Modified Gram–Schmidt orthonormalization. Together, $\widehat{\mathbf{R}}_i$ and $\widehat{\mathbf{t}}_i$ define the predicted keypose $\widehat{k}_i \in \text{SE}(3)$.

B.3 Training Objectives and Implementation Details

Training objectives. The score model is supervised by the segmentation-derived soft mask target $\tilde{w}_m^{(j)} \in [0, 1]$ using a foreground–background balanced binary cross-entropy,

$$\mathcal{L}_{\text{score}}^{(j)} = \text{BCE}_{\text{bal}}\left(\{s_m^{(j)}\}_m, \{\tilde{w}_m^{(j)}\}_m\right),$$

where the two soft-label contributions are normalized separately so that the background does not dominate. This target is used only for training $\mathcal{S}_\theta$ and is absent at inference.

For direct pose regression, let $\mathbf{T}_i^* = (\mathbf{R}_i^*, \mathbf{t}_i^*)$ and $K_j = |\mathcal{I}_j|$. We use

$$\mathcal{L}_{\text{pos}}^{(j)} = \frac{1}{K_j} \sum_{i \in \mathcal{I}_j} \|\widehat{\mathbf{t}}_i - \mathbf{t}_i^*\|_2,$$

$$\mathcal{L}_{\text{mat}}^{(j)} = \frac{1}{9K_j} \sum_{i \in \mathcal{I}_j} \|\widehat{\mathbf{R}}_i - \mathbf{R}_i^*\|_F^2,$$

$$\mathcal{L}_{\text{geo}}^{(j)} = \frac{1}{K_j} \sum_{i \in \mathcal{I}_j} d_{\text{SO}(3)}(\widehat{\mathbf{R}}_i, \mathbf{R}_i^*),$$

$$\mathcal{L}_{\text{offset}}^{(j)} = \frac{1}{K_j} \sum_{i \in \mathcal{I}_j} \|\widehat{\boldsymbol{\delta}}_i - (\mathbf{t}_i^* - \mathbf{c}^{(j)})\|_2.$$

Here $d_{\text{SO}(3)}$ is the standard rotation geodesic distance. For the set $\mathcal{C}_j$ of consecutive valid index pairs, we also supervise the relative transforms:

$$\mathcal{L}_{\text{relative}}^{(j)} = \frac{1}{|\mathcal{C}_j|} \sum_{(i,i^+) \in \mathcal{C}_j} \Big[\|\,\text{trans}(\widehat{\mathbf{T}}_i^{-1} \widehat{\mathbf{T}}_{i^+}) - \text{trans}((\mathbf{T}_i^*)^{-1} \mathbf{T}_{i^+}^*)\|_2 + d_{\text{SO}(3)}\big(\,\text{rot}(\widehat{\mathbf{T}}_i^{-1} \widehat{\mathbf{T}}_{i^+}), \text{rot}((\mathbf{T}_i^*)^{-1} \mathbf{T}_{i^+}^*)\big) \Big],$$

with this term set to zero when $|\mathcal{C}_j| = 0$. The complete pose objective is

$$\mathcal{L}_{\text{pose}}^{(j)} = \lambda_p \mathcal{L}_{\text{pos}}^{(j)} + \lambda_m \mathcal{L}_{\text{mat}}^{(j)} + \lambda_g \mathcal{L}_{\text{geo}}^{(j)} + \lambda_o \mathcal{L}_{\text{offset}}^{(j)} + \lambda_r \mathcal{L}_{\text{relative}}^{(j)},$$

where $(\lambda_p, \lambda_m, \lambda_g, \lambda_o, \lambda_r) = (1.0, 0.2, 0.2, 1.0, 0.1)$; losses are averaged over stages in the batch.

Implementation details. The pose network uses 128 vector channels, five point-level VN blocks, 32 regions, 16 points per region, three relation blocks, four attention heads, and two-layer rotation heads with 256 hidden vector channels. The pose network operates on geometry only; RGB is used solely by the upstream localization module when required. Depending on the number of keypose slots and stage-specific readout heads, the complete model contains approximately 1.21–2.13M parameters across our task configurations. Unless otherwise stated, both stages are optimized with AdamW using a batch size of 1, a constant learning rate of 10^{-4}, and weight decay 10^{-2}; gradients are clipped to a maximum norm of 10. We train the score model for 5,000 epochs and the pose model for 10,000 epochs, freezing $\mathcal{S}_\theta$ throughout pose training. On a single NVIDIA GeForce RTX 4070 with 12 GB of memory, the score and pose stages take approximately 57 minutes and 20–40 minutes, respectively, for our one-demonstration task configurations, corresponding to roughly 1.3–1.6 hours per task in total. Pose-stage augmentation includes random yaw rotations, stage-dependent crop-center jitter, point dropout, partial occlusion, local patch removal, and clipped Gaussian coordinate noise.

C Equivariance Analysis

We analyze equivariant localization, rotation-equivariant prediction on a centered crop, and assembly of the predicted keyposes in the global frame. The operator identities below hold under the stated correspondence assumptions.

C.1 Preliminaries and Group Action

Let $\mathbf{G} = (\mathbf{Q}, \mathbf{b}) \in \mathrm{SE}(3)$, where $\mathbf{Q} \in \mathrm{SO}(3)$, act on a colored point cloud by

$$\mathbf{G} \cdot \mathcal{P} = \{(\mathbf{Q}x_m + \mathbf{b}, f_m)\}_{m=1}^{N}. \tag{16}$$

This is an active transformation of an already reconstructed cloud: appearance features are transported with their points and carry the trivial representation. It is not a new physical camera observation, which can change visibility and appearance. We use $\pi_{\mathbf{G}}$ for the correspondence of indices at each stage of the discrete pipeline.

A type-ℓ feature transforms by $D^{(\ell)}(\mathbf{Q})$; type-0 scalars are invariant and type-1 vectors transform by $\mathbf{v} \mapsto \mathbf{Q}\mathbf{v}$. For C vector channels write $\rho_C(\mathbf{Q}) = \mathbf{I}_C \otimes \mathbf{Q}$. Indexed equivariance means

$$\mathbf{y}'_{\pi_{\mathbf{G}}(m)} = D^{(\ell)}(\mathbf{Q})\mathbf{y}_m. \tag{17}$$

Equivariant maps compose. Invariant scalars may gate equivariant values: if $A'_{\pi(i)\pi(k)} = A_{ik}$ and $\mathbf{y}'_{\pi(k)} = \rho(\mathbf{Q})\mathbf{y}_k$, then

$$\sum_k A'_{\pi(i)\pi(k)}\mathbf{y}'_{\pi(k)} = \rho(\mathbf{Q}) \sum_k A_{ik}\mathbf{y}_k. \tag{18}$$

We assume correspondence-preserving discrete selections throughout: score-based selections have unique outcomes; KNN and farthest-point sampling (FPS) use invariant distances; and the FPS seed and tie breaking preserve point correspondence. Exact ties can instead be excluded.

C.2 Equivariant Score Field and Localization

The score at each corresponding voxel node is a type-0 scalar. The spatial score field still moves with its supporting nodes. We use the score vector $\mathbf{s}^{(j)} = \mathcal{S}_\theta(\mathcal{P}, j)$ defined in Sec. B.1; $\pi_{\mathbf{G}}$ permutes its entries according to voxel-node correspondence.

Proposition 1 (score field and localization). Assume that voxelization and radius-graph construction preserve node and edge correspondence under $\mathbf{G}$, and that support selection and radius cropping preserve correspondence. Then

$$\mathcal{S}_\theta(\mathbf{G} \cdot \mathcal{P}, j) = \pi_{\mathbf{G}}\mathcal{S}_\theta(\mathcal{P}, j), \qquad s^{(j)\prime}_{\pi_{\mathbf{G}}(m)} = s^{(j)}_m, \tag{19}$$

The associated voxel-node coordinates transform by $x^{\mathrm{v}\prime}_{\pi_{\mathbf{G}}(m)} = \mathbf{Q}x^{\mathrm{v}}_m + \mathbf{b}$, so the scores move with their spatial support. If localization $\mathcal{C}(\mathcal{P}, \mathbf{s}^{(j)})$ returns a centered crop $\mathcal{P}^{(j)}_{\mathrm{loc}}$ and center $\mathbf{c}^{(j)}$, then

$$\mathcal{C}(\mathbf{G} \cdot \mathcal{P}, \pi_{\mathbf{G}}\mathbf{s}^{(j)}) = \big(\mathbf{Q} \cdot \mathcal{P}^{(j)}_{\mathrm{loc}}, \mathbf{Q}\mathbf{c}^{(j)} + \mathbf{b}\big). \tag{20}$$

Here $\mathbf{Q}$ rotates centered coordinates and transports their associated appearance features.

Proof. On corresponding graph edges, $r'_{\pi(m)\pi(n)} = \mathbf{Q}r_{mn}$ and $\|r'_{\pi(m)\pi(n)}\|_2 = \|r_{mn}\|_2$. The spherical-harmonic bases and SE(3)-Transformer layers (Fuchs et al., 2020) therefore carry their assigned irreps through the backbone. Its final type-0 channel gives identical scores at corresponding nodes. Masked softmax preserves the scores' correspondence, and deterministic support selection preserves it by assumption. Thus $\widetilde{\alpha}^{(j)\prime}_{\pi_{\mathbf{G}}(m)} = \widetilde{\alpha}^{(j)}_m$, with both sets of weights summing to one. Using the corresponding voxel-node coordinates, we obtain

$$\mathbf{c}^{(j)\prime} = \sum_m \widetilde{\alpha}^{(j)}_m (\mathbf{Q}x^{\mathrm{v}}_m + \mathbf{b}) = \mathbf{Q}\mathbf{c}^{(j)} + \mathbf{b}. \tag{21}$$

Rigid motions preserve distances to this center and thus the corresponding radius crop of the original high-resolution cloud. For every selected point x_m, its centered coordinates transform as $(\mathbf{Q}x_m + \mathbf{b}) - (\mathbf{Q}\mathbf{c}^{(j)} + \mathbf{b}) = \mathbf{Q}(x_m - \mathbf{c}^{(j)})$. This establishes Eq. equation 20. □

C.3 Rotational Equivariance of the Local Pose Model

Centering removes translation from the input action. For a fixed scalar radius $r > 0$, define $\bar{x}_m = (x_m - \mathbf{c}^{(j)})/r$. Proposition 1 yields $\bar{x}'_{\pi_{\mathbf{G}}(m)} = \mathbf{Q}\bar{x}_m$; the local model therefore needs only SO(3) equivariance.

Lemma 1 (point and region features). Assume corresponding crop points, correspondence-preserving KNN and FPS, and distance-based region assignments. Suppose the vector-linear and VN layers preserve type-1 representations. Relabeling corresponding regions by the same g gives

$$\mathbf{H}'_m = \rho_C(\mathbf{Q})\mathbf{H}_m, \quad \mathbf{c}'_g = \mathbf{Q}\mathbf{c}_g, \quad \mathbf{v}^{(0)\prime}_g = \rho_C(\mathbf{Q})\mathbf{v}^{(0)}_g. \tag{22}$$

Proof. Distances preserve KNN and the FPS sequence under the stated seed and tie conditions. For nonzero $\bar{x}_m$, normalization satisfies $\widehat{\mathbf{Q}\bar{x}_m} = \mathbf{Q}\widehat{\bar{x}}_m$; any zero-vector convention must obey the same law. Also $(\mathbf{Q}a) \times (\mathbf{Q}b) = \mathbf{Q}(a \times b)$ for $\mathbf{Q} \in \mathrm{SO}(3)$. Thus all three vector channels of the edge input in Sec. B.2 rotate by $\mathbf{Q}$. The VN layers and masked neighbor/global pooling then yield the stated law for $\mathbf{H}_m$. FPS centers and their nearest-point sets correspond, so $\mathbf{c}'_g = \mathbf{Q}\mathbf{c}_g$ and the pooled vector in Eq. equation 12 transforms by $\rho_C(\mathbf{Q})$. Applying the equivariant ϕ_{reg} proves the token identity in Eq. equation 22. □

Lemma 2 (relation blocks). If attention logits use only invariant scalars, channel norms, distances, and inner products, while value and update maps are type-1 equivariant, then attention weights are invariant and output region tokens remain equivariant.

Proof. By Lemma 1, the channel norms satisfy $\mathbf{u}'_g = \mathbf{u}_g$, and the scalar queries and keys in Sec. B.2 are unchanged. Pairwise squared distances and the channel-wise dot products are also invariant. All three terms of the attention logit therefore satisfy $\ell'_{hgg'} = \ell_{hgg'}$, so both the head-wise softmax and the head average give $A'_{gg'} = A_{gg'}$. These weights gate equivariant values and relative positions:

$$\overline{\mathbf{v}}'_g = \sum_{g'} A_{gg'} \psi_V(\rho_C(\mathbf{Q})\mathbf{v}_{g'}) = \rho_C(\mathbf{Q})\overline{\mathbf{v}}_g,$$
$$\overline{\mathbf{r}}'_g = \sum_{g'} A_{gg'} \mathbf{Q}(\mathbf{c}_{g'} - \mathbf{c}_g) = \mathbf{Q}\overline{\mathbf{r}}_g.$$

The inputs to ϕ_{rel} in Eq. equation 13 consequently carry $2C + 1$ type-1 channels. Its output and the residual both transform by $\rho_C(\mathbf{Q})$, giving $(\mathbf{v}^+_g)' = \rho_C(\mathbf{Q})\mathbf{v}^+_g$. Applying the same argument to each of the three blocks proves the claim. □

Lemma 3 (pose decoding). Let invariant pooling weights aggregate the final region tokens into $\mathbf{z}$. Use the validity mask μ_m, moments $(\boldsymbol{\Sigma}, \boldsymbol{\mu}_3)$, and fusion map of Eqs. equation 14 and equation 15. Suppose the stage-conditioned offset and frame heads consist of equivariant vector operations. Then, for every keypose $i \in \mathcal{I}_j$,

$$\widehat{\boldsymbol{\delta}}'_i = \mathbf{Q}\widehat{\boldsymbol{\delta}}_i, \quad \mathbf{u}'_i = \mathbf{Q}\mathbf{u}_i, \quad \mathbf{v}'_i = \mathbf{Q}\mathbf{v}_i. \tag{23}$$

Proof. Invariant pooling and Lemma 2 give $\mathbf{z}' = \rho_C(\mathbf{Q})\mathbf{z}$. Corresponding masks and N_{valid} are unchanged, while $\bar{x}'_m = \mathbf{Q}\bar{x}_m$ and $\|\bar{x}'_m\| = \|\bar{x}_m\|$. Substitution into Eq. equation 14 gives

$$\boldsymbol{\Sigma}' = \mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^\top, \qquad \boldsymbol{\mu}'_3 = \mathbf{Q}\boldsymbol{\mu}_3.$$

For each channel $\mathbf{z}_c$, therefore, $\boldsymbol{\Sigma}'(\mathbf{Q}\mathbf{z}_c) = \mathbf{Q}(\boldsymbol{\Sigma}\mathbf{z}_c)$. All inputs to ϕ_{mom} are thus type-1 vectors; its VN fusion and the stage-conditioned vector heads preserve this law. The scalar factor 0.45 in the offset head also commutes with $\mathbf{Q}$, and the stage index j is unchanged, yielding Eq. equation 23. □

Proposition 2 (local pose equivariance). Under Lemmas 1–3,

$$\mathcal{F}_\phi(\mathbf{Q} \cdot \mathcal{P}^{(j)}_{\text{loc}}, j) = \mathbf{Q} \cdot \mathcal{F}_\phi(\mathcal{P}^{(j)}_{\text{loc}}, j), \tag{24}$$

where $\mathbf{Q}$ acts on each output vector $(\widehat{\boldsymbol{\delta}}_i, \mathbf{u}_i, \mathbf{v}_i)$.

Proof. The lemmas establish equivariance of the region construction, relation reasoning, and every stage-conditioned readout; composition completes the proof. □

C.4 End-to-End SE(3) Equivariance

The localization center $\mathbf{c}^{(j)}$ both centers the crop and restores the predicted local offset to the global frame:

$$\widehat{\mathbf{t}}_i = \mathbf{c}^{(j)} + \widehat{\boldsymbol{\delta}}_i. \tag{25}$$

Its transformation law follows from Proposition 1. Let $\widehat{\mathbf{R}}_i = \text{IMGS}(\mathbf{u}_i, \mathbf{v}_i)$ denote the rotation readout used by the model. For the end-to-end result, assume that this readout is well-defined on the predicted vectors and obeys

$$\text{IMGS}(\mathbf{Q}\mathbf{u}, \mathbf{Q}\mathbf{v}) = \mathbf{Q}\,\text{IMGS}(\mathbf{u}, \mathbf{v}) \quad \text{for all } \mathbf{Q} \in \text{SO}(3). \tag{26}$$

Any degeneracy handling in the IMGS implementation must preserve this identity. In particular, projection by invariant dot products, normalization, and cross products commute with proper rotations; an iteration built only from these operations inherits the identity. The exact IMGS iteration and its fallback behavior are not specified above, so we keep Eq. equation 26 as an explicit condition.

Corollary 1 (end-to-end equivariance). Under Propositions 1 and 2 and Eq. equation 26, the stage-level predictor $\mathcal{E}_{\theta,\phi}$ satisfies

$$\mathcal{E}_{\theta,\phi}(\mathbf{G} \cdot \mathcal{P}, j) = \mathbf{G} \cdot \mathcal{E}_{\theta,\phi}(\mathcal{P}, j), \qquad \mathbf{G} \cdot (\widehat{\mathbf{T}}_i)_{i \in \mathcal{I}_j} = (\mathbf{G}\widehat{\mathbf{T}}_i)_{i \in \mathcal{I}_j}. \tag{27}$$

Predicted poses are expressed in the input cloud's global frame, so $\text{SE}(3)$ acts on their homogeneous transforms by left multiplication. Here $\widehat{\mathbf{T}}_i$ is the homogeneous-matrix form of the keypose $\widehat{k}_i$ defined in Sec. B.2.

Proof. By Eq. equation 26, $\widehat{\mathbf{R}}'_i = \text{IMGS}(\mathbf{Q}\mathbf{u}_i, \mathbf{Q}\mathbf{v}_i) = \mathbf{Q}\widehat{\mathbf{R}}_i$. For the translation,

$$\widehat{\mathbf{t}}'_i = (\mathbf{Q}\mathbf{c}^{(j)} + \mathbf{b}) + \mathbf{Q}\widehat{\boldsymbol{\delta}}_i = \mathbf{Q}\widehat{\mathbf{t}}_i + \mathbf{b}.$$

Thus

$$\widehat{\mathbf{T}}'_i = \begin{bmatrix} \mathbf{Q}\widehat{\mathbf{R}}_i & \mathbf{Q}\widehat{\mathbf{t}}_i + \mathbf{b} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \mathbf{G}\widehat{\mathbf{T}}_i \tag{28}$$

for every index in the ordered subsequence. □

Equivariance in practice. The preceding analysis establishes the SE(3) transformation law for EquiKey when a reconstructed point cloud and its induced neighborhoods, crops, and regions maintain their correspondences. In practice, voxelization and point sampling may perturb these correspondences, while a change in camera viewpoint may alter visibility and depth. The analysis identifies the geometric structure preserved by the model under these well-defined conditions, with practical deviations arising during input construction. This design remains effective in our settings: EquiKey generalizes to unseen spatial configurations from a single demonstration, as shown in our experiments.

Table 4: **EQUIKEY-VLA Architecture Specifications.** Rows marked with ($\star$) highlight our proposed SE(3)-equivariant keypose conditioning mechanisms.

Module	Layer Type	Layer Num	Input Shape	Output Shape
SigLIP Vision Tower (per view, $V = 3$)				
SiglipVisionTransformer	SiglipVisionEmbeddings	1	$(B, 3, 224, 224)$	$(B, 256, 1152)$
	SiglipEncoderLayer	27	$(B, 256, 1152)$	$(B, 256, 1152)$
	LayerNorm	1	$(B, 256, 1152)$	$(B, 256, 1152)$
PaliGemma Multi-Modal Projector (per view)				
Vision Projector	Linear	1	$(B, 256, 1152)$	$(B, 256, 2048)$
Understanding Expert (Gemma-2B)				
Text Embedding ($\star$)	Embedding	1	$(B, 320)$	$(B, 320, 2048)$
Language Transformer ($\star$)	GemmaDecoderLayer	18	$(B, 1088, 2048)$	$(B, 1088, 2048)$
Norm	RMSNorm	1	$(B, 1088, 2048)$	$(B, 1088, 2048)$
Action Expert (Gemma-300M)				
Expert Transformer ($\star$)	GemmaDecoderLayer	18	$(B, 30, 1024)$	$(B, 30, 1024)$
Norm	AdaRMSNorm	1	$(B, 30, 1024)$	$(B, 30, 1024)$
Action Output Projection	Linear	1	$(B, 30, 1024)$	$(B, 30, 32)$
Projection / Time Embedding				
Time Embedding	Sin/Cos	1	(B)	$(B, 1024)$
Keypose Projection E_k ($\star$)	Linear + GeLU + Linear(zero-init)	1	$(B, 32)$	$(B, 1024)$
Action In Projection	Linear	1	$(B, 30, 32)$	$(B, 30, 1024)$

D EQUIKEY-VLA ARCHITECTURE

EQUIKEY-VLA builds on the $\pi_{0.5}$ Mixture-of-Transformers (MoT) backbone (Black et al., 2025). We investigate keypose conditioning through the Understanding Expert, the Action Expert, and both via dual-stream conditioning. Our primary configuration conditions only the Understanding Expert, as selected by the ablation study. The 20-dimensional robot state, action, and keypose representations are zero-padded to match the 32-dimensional interface of the pretrained $\pi_{0.5}$.

Understanding Expert Conditioning. The VLM extracts spatial-semantic features from visual and textual inputs. Three camera views ($V = 3$) are processed via a SigLIP-So400M vision tower (27 SiglipEncoderLayers with hidden dimension 1152), producing 768 visual tokens, which are subsequently mapped to a 2048-dimensional feature space via a linear Vision Projector. For textual input, in addition to task instructions ℓ_τ and current robot state s_t, we inject the normalized keypose $k_t \in \mathbb{R}^{32}$. Each dimension of k_t is normalized using q01/q99 quantile statistics and discretized into 256 uniform bins. The serialized keypose tokens are appended directly to the text prompt template:

"Task:{ℓ_τ}, State:{s_t}, Keypose:{k_t}; Action:"

The resulting text prompt sequence (padded to 320 tokens) is embedded into a 2048-dimensional representation via the shared Text Embedding layer and concatenated with the 768 visual tokens, yielding a total sequence length of 1088 tokens for the 18-layer Gemma-2B decoder.

Action Expert AdaRMSNorm Modulation. The Action Expert generates continuous action trajectories $a_{t:t+H} \in \mathbb{R}^{30\times 32}$ ($H = 30$) via flow matching. To inject continuous geometric guidance, the normalized keypose $\bar{k}_t$ is projected into a 1024-dimensional conditioning vector via a two-layer MLP E_k. The validity-masked keypose embedding $m_t^k E_k(\bar{k}_t)$ is added directly to the sinusoidal flow-step time embedding $e_\lambda \in \mathbb{R}^{1024}$ to form the unified conditioning vector $\tilde{e}_{\lambda,t} \in \mathbb{R}^{1024}$. This vector modulates the AdaRMSNorm layers across the 18 GemmaDecoderLayers, steering the denoising trajectory according to the geometric keypose condition. In the dual-stream variant, both the textual keypose conditioning in the Understanding Expert and the AdaRMSNorm modulation in the Action Expert are enabled simultaneously.

E RoboTwin 2.0 Simulation Details



Figure 6: **Some RoboTwin 2.0 tasks and predicted keypose sequences.** The top and bottom rows show the clean and randomized settings, respectively, with colored coordinate frames denoting the nearest future SE(3) keypose used for current guidance.

Task Split. Following Li et al. (2026a), we categorize the 50 tasks according to their manipulation horizons, including 30 tasks with $H = 1$, 15 tasks with $H = 2$, and 5 tasks with $H = 3$. We randomly select 10 tasks as the task out-of-distribution (OOD) split while approximately preserving the overall horizon distribution, and use the remaining 40 tasks as the in-distribution (ID) split. Some $H = 3$ RoboTwin tasks and their predicted keypose sequences

VLA Pretraining Details. The shared EquiKey-VLA backbone is pretrained exclusively on the 40 ID tasks; none of the 10 OOD-task demonstrations are used during its pretraining. For π_0 and $\pi_{0.5}$, we use the mixed training set described in the main text. Unless otherwise stated, these models use the same training configuration. All models are trained using JAX with fully sharded data parallel (FSDP) training across 8 GPUs, each with 80 GB of memory. We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.95$, a learning rate of 5×10^{-5}, and a warm-up of 100 optimization steps. Each GPU uses a

Table 5: **Per-task comparison with one-shot adaptation baselines on RoboTwin 2.0.** We report success rates (%) for the 10 OOD tasks under the Easy and Hard settings.

Simulation Task	Horizon	$\pi_{0.5}$ + FLA		$\pi_{0.5}$ + FTM		$\pi_{0.5}$ + LoRA		**EquiKey-VLA**	
		Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	1	42%	36%	46%	34%	43%	31%	**98%**	**74%**
Blocks Ranking Size	3	24%	21%	29%	27%	37%	24%	**50%**	**46%**
Click Bell	1	**100%**	91%	**100%**	**100%**	95%	**100%**	**100%**	92%
Move Stapler Pad	1	0%	9%	5%	13%	12%	6%	**62%**	**57%**
Pick Dual Bottles	2	77%	84%	**92%**	62%	88%	87%	86%	**97%**
Place A2B Right	1	9%	42%	34%	44%	34%	28%	**81%**	**85%**
Place Bread Basket	1	14%	35%	27%	21%	31%	29%	**71%**	**62%**
Place Bread Skillet	2	0%	0%	0%	5%	36%	5%	**63%**	**64%**
Place Cans Plasticbox	2	0%	0%	0%	6%	30%	23%	**61%**	**95%**
Shake Bottle Horizontally	1	**100%**	**100%**	99%	95%	**100%**	**100%**	99%	97%
Average OOD Tasks (%)	–	36.60	41.80	43.20	40.70	50.60	43.30	**77.10**	**76.90**

batch size of 32, resulting in a global batch size of 256. We clip the gradient norm to 1.0 and maintain an exponential moving average (EMA) of the model parameters with a decay rate of 0.99. These models are trained for 40,001 optimization steps. For X-VLA, we use its official default training configuration with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$) and a learning rate of 1×10^{-4}. The learning rate is warmed up for 2,000 steps, while the VLM and Transformer core are frozen for the first 1,000 steps. Training is performed on the same 8-GPU setup with a batch size of 16 per GPU and gradient accumulation over 2 steps, resulting in an effective global batch size of 256. We clip the gradient norm to 1.0 and train X-VLA for 40,000 optimization steps.

One-Shot Adaptation Baselines. Following Li et al. (2026b), we adapt the pretrained $\pi_{0.5}$ checkpoint with three parameter-efficient baselines. **LoRA** follows the official $\pi_{0.5}$ recipe, inserting low-rank adapters into the attention and MLP projections of both Gemma experts (r=16 for the 2B VLM and r=32 for the 300M action expert), while keeping the SigLIP encoder and action/time projection layers trainable, resulting in 467M trainable parameters. **FLA** adapts only the visual encoder by inserting rank-16 LoRA adapters into the MLP layers of all 27 SigLIP blocks, resulting in 4.7M trainable parameters. **FTM** freezes the entire VLA and learns only a channel-wise affine modulation, $\hat{F} = (1 + \gamma) \odot F + \beta$, on the SigLIP output tokens, with 4k trainable parameters. All adapters are initialized to preserve the pretrained model at the start of adaptation.

All three baselines share the same training configuration. For each OOD task, the adaptor is trained independently using a single randomly sampled demonstration from that task. Training is performed in JAX on a single 80 GB GPU with a batch size of 16. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 10^{-10}, and gradient-norm clipping at 1.0. The learning rate is linearly warmed up for 100 steps and then decayed using a cosine schedule, with a peak learning rate of 2.5×10^{-5} for LoRA and FLA and 1×10^{-3} for FTM. Frozen parameters are stored in bfloat16 and no EMA is used, following the official $\pi_{0.5}$ LoRA setting. Each task-specific adaptor is trained for 1,000 optimization steps, and we evaluate the final checkpoint. Detailed training configurations of EQUIKEY are provided in Appendix B.3.

Detailed Per-Task Results. We provide detailed per-task results for all simulation experiments. Table 5 compares task-specific one-shot adaptation methods, while Table 6 reports the results of general VLA baselines on the 10 OOD tasks. Tables 8 and 9 provide the corresponding per-task results for the keypose-conditioning and scheduling/control ablations, respectively. For each OOD task, we evaluate both the Easy setting without perturbations to lighting, background, or table height and the Hard setting with stronger variations in these factors. For completeness, per-task results on the 40 ID tasks are reported in Table 7.

Table 6: **Per-task comparison with general VLA baselines on RoboTwin 2.0.** We report success rates (%) for the 10 OOD tasks under the Easy and Hard settings.

Simulation Task	Horizon	π_0		$\pi_{0.5}$		X-VLA		**EquiKey-VLA**	
		Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	1	8%	2%	14%	13%	7%	1%	**98%**	**74%**
Blocks Ranking Size	3	26%	16%	11%	9%	23%	24%	**50%**	**46%**
Click Bell	1	97%	95%	89%	**96%**	90%	89%	**100%**	92%
Move Stapler Pad	1	0%	0%	5%	3%	0%	1%	**62%**	**57%**
Pick Dual Bottles	2	25%	18%	**89%**	62%	53%	51%	86%	**97%**
Place A2B Right	1	19%	24%	33%	34%	17%	16%	**81%**	**85%**
Place Bread Basket	1	0%	1%	4%	3%	13%	9%	**71%**	**62%**
Place Bread Skillet	2	0%	0%	1%	1%	1%	3%	**63%**	**64%**
Place Cans Plasticbox	2	0%	0%	1%	2%	2%	0%	**61%**	**95%**
Shake Bottle Horizontally	1	96%	88%	**99%**	**100%**	97%	98%	**99%**	97%
Average OOD Tasks (%)	–	27.10	24.40	34.60	32.30	30.30	29.20	**77.10**	**76.90**

Table 7: **Per-task results of general VLA baselines on the 40 ID tasks of RoboTwin 2.0.** We report success rates (%) for the 40 ID tasks under the Easy and Hard settings.

Simulation Task	Horizon	π_0		$\pi_{0.5}$		X-VLA	
		Easy	Hard	Easy	Hard	Easy	Hard
Beat Block Hammer	1	96%	87%	94%	93%	95%	86%
Blocks Ranking RGB	3	20%	55%	96%	92%	95%	92%
Click Alarmclock	1	98%	96%	99%	98%	3%	7%
Dump Bin Bigbin	1	78%	66%	85%	84%	54%	58%
Grab Roller	1	96%	96%	100%	99%	96%	94%
Handover Block	2	19%	8%	78%	35%	72%	62%
Handover Mic	2	94%	99%	94%	96%	99%	99%
Hanging Mug	2	14%	13%	39%	28%	48%	46%
Lift Pot	1	55%	62%	88%	83%	96%	94%
Move Can Pot	1	85%	72%	81%	83%	92%	83%
Move Pillbottle Pad	1	54%	40%	87%	76%	92%	79%
Move Playingcard Away	1	71%	56%	93%	86%	82%	83%
Open Laptop	1	86%	87%	93%	96%	90%	93%
Open Microwave	1	59%	67%	53%	63%	37%	49%
Pick Diverse Bottles	2	27%	18%	81%	64%	57%	47%
Place A2B Left	1	50%	36%	91%	85%	42%	45%
Place Burger Fries	2	38%	32%	88%	79%	97%	72%
Place Can Basket	2	56%	58%	77%	67%	84%	61%
Place Container Plate	1	96%	92%	94%	93%	94%	99%
Place Dual Shoes	2	71%	76%	71%	62%	81%	81%
Place Empty Cup	1	96%	87%	100%	96%	90%	88%
Place Fan	1	67%	68%	80%	85%	72%	73%
Place Mouse Pad	1	23%	16%	41%	41%	53%	52%
Place Object Basket	2	83%	79%	77%	79%	77%	71%
Place Object Scale	1	56%	49%	88%	81%	79%	69%
Place Object Stand	1	90%	83%	100%	98%	91%	84%
Place Phone Stand	1	52%	35%	78%	75%	88%	82%
Place Shoe	1	88%	79%	100%	99%	96%	92%
Press Stapler	1	56%	47%	80%	81%	62%	54%
Put Bottles Dustbin	3	54%	66%	74%	80%	74%	77%
Put Object Cabinet	2	42%	32%	76%	64%	63%	48%
Rotate QRcode	1	92%	85%	100%	100%	84%	87%
Scan Object	2	37%	41%	63%	64%	55%	47%
Shake Bottle	1	100%	99%	100%	99%	99%	97%
Stack Blocks Three	3	65%	49%	84%	72%	68%	53%
Stack Blocks Two	2	74%	64%	100%	93%	94%	90%
Stack Bowls Three	3	60%	58%	79%	72%	73%	72%
Stack Bowls Two	2	85%	85%	93%	95%	92%	91%
Stamp Seal	1	68%	57%	78%	72%	61%	50%
Turn Switch	1	60%	64%	83%	87%	55%	54%
Average ID Tasks (%)	–	65.28	61.48	83.90	79.88	75.80	71.53

Table 8: **Per-task ablation of keypose guidance and conditioning on RoboTwin 2.0.** We report success rates (%) on the 10 OOD tasks under the Easy and Hard settings.

Simulation Task	Horizon	Oracle		**Ours**		Dual Stream		Action Expert		w/o Keypose		w/o Dropout	
		Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	1	94%	86%	98%	**74%**	**99%**	72%	91%	68%	14%	13%	91%	71%
Blocks Ranking Size	3	63%	59%	**50%**	**46%**	44%	45%	31%	38%	11%	9%	29%	24%
Click Bell	1	100%	100%	**100%**	92%	**100%**	90%	**100%**	89%	89%	**96%**	**100%**	82%
Move Stapler Pad	1	98%	93%	**62%**	**57%**	53%	38%	37%	45%	5%	3%	54%	45%
Pick Dual Bottles	2	89%	89%	86%	97%	85%	99%	68%	**100%**	89%	62%	**90%**	91%
Place A2B Right	1	94%	92%	81%	85%	82%	**86%**	80%	67%	33%	34%	**84%**	77%
Place Bread Basket	1	89%	97%	**71%**	62%	66%	**64%**	52%	57%	4%	3%	67%	58%
Place Bread Skillet	2	88%	86%	**63%**	64%	62%	67%	35%	37%	1%	1%	56%	**71%**
Place Cans Plasticbox	2	44%	41%	61%	**95%**	57%	**95%**	17%	14%	1%	2%	**64%**	**95%**
Shake Bottle Horizontally	1	98%	98%	**99%**	97%	90%	98%	86%	99%	**99%**	**100%**	92%	97%
Average OOD Tasks	–	85.70	84.10	**77.10**	**76.90**	73.80	75.40	59.70	61.40	34.60	32.30	72.70	71.10

Table 9: **Per-task ablation of keypose scheduling and control on RoboTwin 2.0.** We report success rates (%) on the 10 OOD tasks of RoboTwin 2.0 under the Easy and Hard settings.

Simulation Task	Horizon	**Ours**		Reach Scheduler		Interpolation Planning	
		Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	1	**98%**	74%	96%	**77%**	93%	72%
Blocks Ranking Size	3	**50%**	**46%**	35%	30%	3%	2%
Click Bell	1	**100%**	**92%**	**100%**	**92%**	86%	62%
Move Stapler Pad	1	**62%**	**57%**	54%	52%	19%	0%
Pick Dual Bottles	2	86%	**97%**	62%	63%	**87%**	91%
Place A2B Right	1	**81%**	**85%**	73%	74%	45%	83%
Place Bread Basket	1	**71%**	**62%**	11%	25%	0%	0%
Place Bread Skillet	2	63%	**64%**	42%	48%	**68%**	48%
Place Cans Plasticbox	2	61%	**95%**	53%	41%	**62%**	77%
Shake Bottle Horizontally	1	**99%**	97%	87%	88%	93%	**100%**
Average OOD Tasks	–	**77.10**	**76.90**	61.30	59.00	55.60	53.50

E.1 Ablation Details

Reach Scheduler. For this ablation, we retain the same predicted keyposes and VLA policy but replace the proposed time-based scheduler with a geometric switching criterion. The current keypose is advanced once the end-effector of every active arm is within 2 cm of its corresponding target position; orientation is not used for switching.

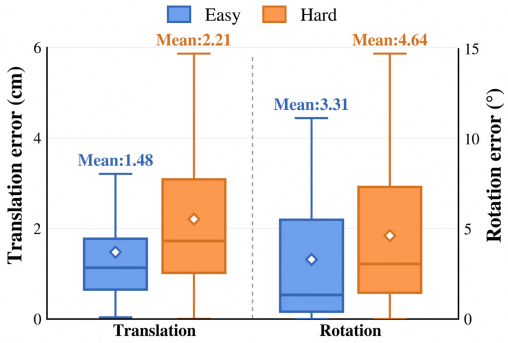


Figure 7: **Keypose prediction errors on the 10 OOD tasks of RoboTwin 2.0.**

Interpolation Planning. For this ablation, we remove the VLA policy and directly execute the predicted keypose sequence using deterministic $\mathrm{SE}(3)$ interpolation. For each segment, the number of interpolation steps is $n = \min\{120, \max(\lceil d_p/1\,\mathrm{cm} \rceil, \lceil d_R/15^\circ \rceil, 1)\}$, where d_p and d_R denote the translational and rotational distances, respectively. Dual-arm segments use the larger step count required by either arm for synchronized execution. Positions are linearly interpolated, while orientations use shortest-path quaternion SLERP.

Keypose Prediction Accuracy. We additionally evaluate the geometric accuracy of the keyposes predicted by EquiKey on the 10 OOD tasks of RoboTwin 2.0. As shown in Fig. 7, the mean translation errors are 1.48 cm and 2.21 cm under the Easy and Hard settings, respectively, while the corresponding mean rotation errors are 3.31° and 4.64°. These results show that the predicted keyposes remain geometrically accurate under both evaluation settings.

F Real-World Setup and Task Details

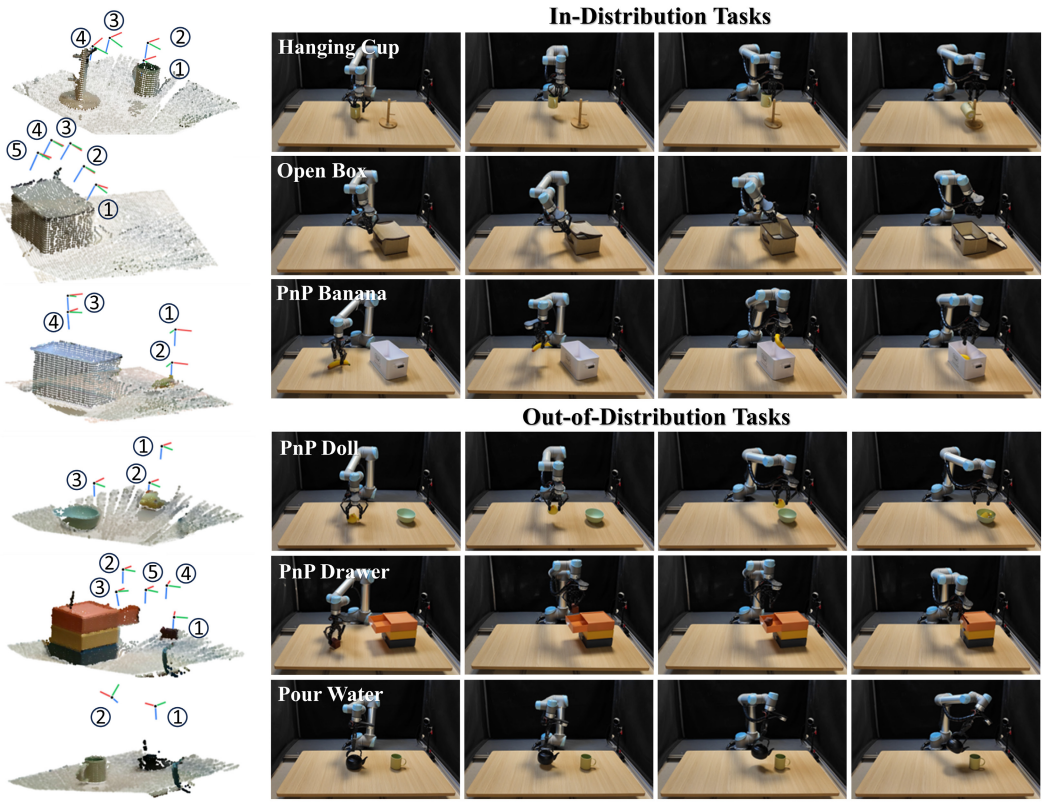


Figure 8: **Real-world tasks.** The top three rows correspond to the ID tasks, while the bottom three rows show the OOD tasks. The left column visualizes the predicted EquiKey keypose sequence for each task, and the remaining columns show the task execution process.

Hardware and Control Setup. Our real-world platform consists of a 6-DoF UR5 arm equipped with a parallel-jaw gripper and two RealSense L515 cameras, including a global camera and a wrist-mounted camera. For data collection, we use a 3Dconnexion SpaceMouse for teleoperation. The SpaceMouse commands are sent to the robot through Cartesian velocity control at 15 Hz, with a 7-dimensional action space defined as $a_t = [\Delta x, \Delta y, \Delta z, \Delta\theta_x, \Delta\theta_y, \Delta\theta_z, g_t] \in \mathbb{R}^6 \times [0, 1]$. During policy inference, the UR5 is controlled through the RTDE interface using Cartesian end-effector position control at 15 Hz. We set the action chunk size to 30 and employ RTC (Black et al., 2026) for smoother execution. The complete model inference and robot control pipeline runs on a local PC equipped with an Intel Core i7-12700KF 3.6 GHz CPU, 32 GB RAM, and an NVIDIA RTX 4070 GPU. Hand–eye calibration between the camera and robot is performed using a checkerboard. The resulting extrinsic transformation is then used to transform the point cloud from the global camera frame into the robot base frame.

Task Details. Demonstrations are collected via teleoperation at 15 Hz. We collect 50 demonstrations for each of the 3 ID tasks (150 in total), and only one demonstration for each of the 3 OOD tasks (3 in total), as shown in Fig. 8. All demonstrations are collected within the Area-ID region illustrated in Fig. 4. The task instructions and average trajectory lengths are detailed below.

- *Hanging Cup:* "Pick up the cup and hang it on the angled rack." (632 steps)
- *Open Box:* "Open the lid of the canvas box." (297 steps)
- *PnP Banana:* "Pick up the banana and put it in the white box." (610 steps)
- *PnP Doll:* "Move the yellow doll into the green bowl." (583 steps)
- *PnP Drawer:* "Place the block inside the drawer and then push the drawer closed." (702 steps)
- *Pour Water:* "Grasp the black teapot, tilt it, and pour water into the cup." (571 steps)

Table 10: **Per-task real-world evaluation results.** Success rates are reported for each real-world task. For ID tasks, we report the pooled results over 20 rollouts, together with separate results for the Area-ID and Area-OOD settings, with 10 rollouts in each setting. For OOD tasks, results are pooled over 20 rollouts per task.

Task Name	π_0	$\pi_{0.5}$	X-VLA	**Ours**
In-Distribution (ID) Tasks				
Hanging Cup	45.0	50.0	35.0	65.0
+ Area-ID	90.0	100.0	70.0	80.0
+ Area-OOD	0.0	0.0	0.0	50.0
Open Box	60.0	60.0	40.0	95.0
+ Area-ID	100.0	90.0	80.0	100.0
+ Area-OOD	20.0	30.0	0.0	90.0
PnP Banana	40.0	45.0	35.0	75.0
+ Area-ID	80.0	90.0	70.0	90.0
+ Area-OOD	0.0	0.0	0.0	60.0
Average ID	48.3	<u>51.7</u>	36.7	**78.3**
+ Area-ID	<u>90.0</u>	**93.3**	73.3	<u>90.0</u>
+ Area-OOD	6.7	<u>10.0</u>	0.0	**66.7**
Out-of-Distribution (OOD) Tasks				
PnP Doll	20.0	25.0	10.0	70.0
PnP Drawer	10.0	10.0	0.0	75.0
Pour Water	0.0	0.0	0.0	75.0
Average OOD	10.0	<u>11.7</u>	3.3	**73.3**