SURE: SPATIOTEMPORAL UNCERTAINTY REPRESENTATION FOR EFFICIENT VISION-LANGUAGE-ACTION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies have become a promising paradigm for end-to-end robot control, but their effectiveness still depends heavily on large volumes of demonstration data. In manipulation tasks, this dependence is compounded by structured spatiotemporal uncertainty in the action space: multiple nearby actions or trajectories may successfully solve the same task, while semantically equivalent demonstrations often unfold at different speeds. Standard pointwise supervision treats each demonstrated action as an exact target, imposing an unnecessary learning burden under such uncertainty. We present SURE, a data-efficient VLA training framework that explicitly models both spatial and temporal uncertainty. SURE augments a VLA backbone with a lightweight transposed-convolutional probabilistic action decoder that captures spatial uncertainty through local correlations among neighboring action bins. It further introduces Soft-DTW alignment supervision to accommodate local timing variations between predicted and demonstrated action chunks. Together, these two components reduce sensitivity to demonstrator-specific poses and execution timing, enabling more data-efficient action learning. We evaluate SURE on LIBERO, RoboTwin 2.0, and real-world manipulation tasks. Even with only 10% of the demonstration data or 20% of the training steps, SURE achieves manipulation success rates that match or surpass baseline VLAs.

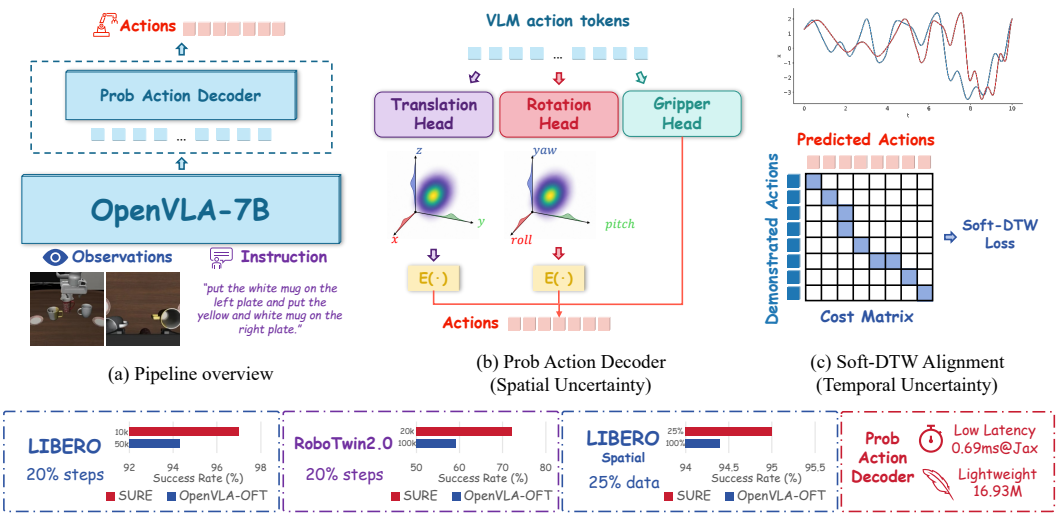
1 INTRODUCTION

Embodied AI and robot foundation models have advanced rapidly in recent years, driven by the emergence of large-scale vision-language-action (VLA) policies that directly map visual observations and language instructions to low-level control (Brohan et al., 2023; Ghosh et al., 2024; Kim et al., 2025b; Black et al., 2025). As a result, a central question in robot learning is no longer whether such end-to-end policies can work, but which action-learning paradigm can most efficiently turn expensive robot demonstrations into robust manipulation behaviors. This question is important in practical settings, where collecting real-world robot data remains slow, costly, and inherently noisy.

One source of difficulty is the inherent uncertainty in robot action data. Many manipulation tasks admit multiple valid solutions which the end effector can approach from slightly different poses, move at different speeds, or follow different but equally successful trajectories. Importantly, this uncertainty is structured rather than arbitrary. It typically exhibits strong local spatiotemporal correlations, actions that are close to a successful action in action space, or adjacent in time to a successful motion pattern, are often still correct. Treating supervision as a single exact target therefore introduces an unnecessary learning burden. A policy can be semantically correct yet still incur a large penalty simply because its prediction does not match the demonstrator's exact pose or timing.

Existing action representations only partially address this issue. Early and widely used VLA decoders often rely on tokenized or autoregressive action prediction (Brohan et al., 2023; Kim et al., 2025b), while recent systems increasingly adopt diffusion- and flow-based action heads to better model multimodal futures (Chi et al., 2023; Black et al., 2025; NVIDIA et al., 2025). These generative action heads are expressive and have become a popular choice for modern VLAs. However, their flexibility does not automatically yield data efficiency. When supervision still treats each ac-

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.



(a) Pipeline overview (b) Prob Action Decoder (Spatial Uncertainty) (c) Soft-DTW Alignment (Temporal Uncertainty)

Figure 1: Overview of SURE. (a) Given observations and a language instruction, an OpenVLA-7B backbone produces action tokens that are decoded into action chunks. (b) Our probabilistic action decoder models translation and rotation as discretized distributions and recovers continuous controls with expectation-based decoding, capturing spatial uncertainty and multimodality. (c) During training, Soft-DTW alignment supervision aligns predicted and demonstrated action chunks that progress at different speeds, addressing temporal uncertainty in robot demonstrations.

tion label as an isolated point, the model must discover from scratch that neighboring actions and slightly shifted trajectories should receive similar credit. In practice, this often means that the policy needs more data before it can internalize the local structure of valid robot behavior.

Recent work suggests that structured probabilistic action representations can be more sample-efficient and interpretable than unstructured point prediction. Instead of regressing a single action target, these methods model uncertainty with discretized action spaces, heatmap-like predictions, or voxel-space probability distributions, thereby injecting locality and multimodality into the output space (James et al., 2022; Shridhar et al., 2023; Goyal et al., 2023; Yang et al., 2026). However, most of this literature focuses primarily on spatial uncertainty, while temporal uncertainty is still typically handled with exact per-step supervision.

Temporal uncertainty creates a second challenge. Even for the same task and intent, demonstrations can unfold at different speeds because of operator preferences, hesitation, embodiment dynamics, or minor contact events. Standard framewise regression losses such as L1 or L2 fail to account for this uncertainty and instead penalize timing differences directly. As a result, the model may overfit to demonstrator-specific execution speeds rather than the underlying behavior. This problem is closely related to the alignment problem long studied in speech recognition, where dynamic time warping (DTW) was introduced to compare signals that differ in local speaking rate (Sakoe & Chiba, 1978). Its differentiable variant, Soft-DTW (Cuturi & Blondel, 2017), offers a natural way to bring this idea into gradient-based policy learning, yet it remains underexplored as a supervision mechanism for action prediction in VLA-style robot policies.

In this paper, we tackle both spatial and temporal uncertainty with a simple, data-efficient VLA design, as shown in Figure 1. We introduce a lightweight probabilistic action decoder built with transposed convolutions that models spatial uncertainty through local correlations among neighboring action bins and can be supervised directly with standard pointwise losses. We further couple this head with Soft-DTW alignment supervision, which aligns action sequences executed at different speeds and accounts for temporal uncertainty across demonstrations. Together, these design choices encourage the policy to learn a distribution over valid behaviors rather than memorize a one-to-one mapping from observations to exact action labels. Empirically, SURE outperforms strong baselines with matched or fewer optimization steps. On LIBERO(Liu et al., 2023), our method reaches 97.0% average success after 10k steps, surpassing OpenVLA-OFT trained for 50k steps. With only 25%

of the LIBERO-Spatial data, our method also exceeds the full-data baseline. These gains extend to RoboTwin 2.0 (Chen et al., 2026) and our real-world benchmark.

Our contributions are threefold. First, we introduce a lightweight transposed-convolutional probabilistic action decoder that models spatial uncertainty with locally correlated action bins while enabling fast, parallel expectation-based decoding without iterative denoising. Second, we incorporate Soft-DTW alignment supervision to improve robustness to demonstrations with similar task intent but different execution speeds. Third, extensive experiments on LIBERO, RoboTwin 2.0, and our real-world benchmark show that our method achieves stronger training and data efficiency than competitive baselines, requiring fewer optimization steps or less training data.

2 Related Work

Embodied AI Large-scale embodied AI is currently pursued through two complementary paradigms. Vision-language-action (VLA) policies directly map observations and language instructions to executable actions, typically benefiting from large, heterogeneous robot datasets and pretrained multimodal backbones (Brohan et al., 2023; Open X-Embodiment Collaboration et al., 2024; Ghosh et al., 2024; Kim et al., 2025b; Black et al., 2025; NVIDIA et al., 2025). In parallel, world-model and world-action-model approaches augment policy learning with future prediction, imagination, or simulation, enabling planning, policy evaluation, or synthetic data generation in learned latent or pixel spaces (Hafner et al., 2020; Liao et al., 2026; Guo et al., 2026; Zhou et al., 2026).

Action Representation Action representation is a central design choice in robot policy learning. Early VLA and imitation-learning systems commonly relied on discretized or autoregressive action prediction, where continuous control is converted into token sequences or predicted one step at a time (Brohan et al., 2023; Kim et al., 2025b; Pertsch et al., 2025). More recent work has explored continuous parallel decoding and improved fine-tuning recipes to reduce inference cost and increase adaptation performance (Kim et al., 2025a; Wen et al., 2025). Another major direction models actions as distributions rather than single point estimates. This includes diffusion- and flow-based action heads for multimodal action generation (Chi et al., 2023; Black et al., 2025; NVIDIA et al., 2025), as well as structured probabilistic outputs such as discretized spatial bins, heatmaps, and voxel-space distributions (James et al., 2022; Shridhar et al., 2023; Goyal et al., 2023; Yang et al., 2026). Our method is most closely related to the latter line of work. Specifically, we use a transposed-convolutional head to explicitly model the action probability distribution and its decoding process. We optimize the decoder primarily with L1 action supervision and use only a lightly weighted auxiliary distribution loss, rather than relying on dense distributional supervision as the primary learning signal.

DTW-Based Temporal Alignment Dynamic time warping (DTW) has long been a standard tool for aligning sequences with non-uniform temporal progress (Sakoe & Chiba, 1978). Soft-DTW extends this idea to a differentiable objective, making it suitable for gradient-based learning (Cuturi & Blondel, 2017). In robotics, DTW-style alignment has been used for demonstration comparison, weak supervision, and imitation under temporal or action mismatches (Yeh et al., 2025). These works highlight the practical importance of alignment under variable execution speeds when different trajectories express the same underlying intent. Rather than using DTW only for retrieval, comparison, or offline alignment, we use Soft-DTW alignment supervision for VLA action learning, allowing the policy to learn end-to-end from temporally misaligned but semantically consistent demonstrations.

Data-Efficient Robot Learning. Improving data efficiency remains a core challenge in robot learning because large-scale demonstrations are expensive to collect, especially on real hardware. Existing efforts improve efficiency from several angles: scaling diverse cross-embodiment datasets for transfer (Open X-Embodiment Collaboration et al., 2024; Ghosh et al., 2024; Kim et al., 2025b), designing better adaptation and fine-tuning recipes (Kim et al., 2025a), building smaller or faster VLA architectures (Wen et al., 2025), and injecting stronger geometric structure into the action output space (James et al., 2022; Shridhar et al., 2023; Goyal et al., 2023; Yang et al., 2026). Our approach is complementary to these directions. Rather than relying on more pretraining data, larger-scale models, or strongly weighted target-distribution supervision, we improve data efficiency through a joint design of structured action representation and temporally tolerant supervision.

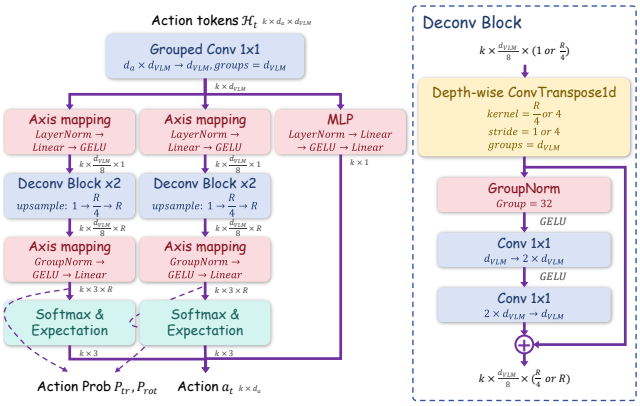


Figure 2: Architecture of the probabilistic action decoder. Action tokens $\mathcal{H}_t \in \mathbb{R}^{k \times d_a \times d_{VLM}}$ are first fused by a grouped 1×1 convolution and then split into translation, rotation, and gripper branches. Translation and rotation use two Deconv Blocks to upsample an ordered bin axis from $1 \to R/4 \to R$. The resulting per-axis logits over R bins are normalized by softmax and decoded into continuous actions by expectation. Each Deconv Block combines depthwise transposed convolution with pointwise residual layers to encode local correlations among neighboring bins. The gripper action is decoded by a lightweight MLP.

3 METHODS

Our method builds on an OpenVLA-7B backbone and introduces two uncertainty-aware components for data-efficient action learning. The probabilistic action decoder represents spatial uncertainty in translation and rotation, while Soft-DTW alignment supervision accounts for temporal uncertainty among semantically consistent action sequences. At time step t, the policy takes a visual observation $\mathcal{I}_t$, proprioceptive state p_t, and language instruction $\mathcal{T}$ as input. The VLA backbone produces a chunk of action tokens

$$\mathcal{H}_t = \mathrm{VLM}(\mathcal{I}_t, p_t, \mathcal{T}) \in \mathbb{R}^{k \times d_a \times d_{VLM}}, \tag{1}$$

where k is the action chunk length, d_a is the action dimension, and d_{VLM} is the hidden size of the backbone. The probabilistic action decoder maps these tokens to an action chunk $\mathcal{A}_t \in \mathbb{R}^{k \times d_a}$. Soft-DTW alignment supervision then aligns predicted and demonstrated action chunks whose trajectories are semantically consistent but differ in local execution rate.

3.1 PROBABILISTIC ACTION DECODER

Robot manipulation rarely admits a unique action target. Even under the same observation and instruction, multiple nearby end-effector poses, orientations, or contact timings can all lead to success. This makes the action space both locally correlated and potentially multimodal. Direct scalar regression ignores this structure and forces the model to match one exact demonstrator trajectory.

For each continuous translation and rotation dimension, we discretize the normalized action range $[-1, 1]$ into $R = 256$ ordered bins with centers $\{c_r\}_{r=1}^{R}$. As illustrated in Figure 2, the decoder first compresses $\mathcal{H}_t$ with a grouped convolution and then sends the resulting features to translation, rotation, and gripper branches. The translation and rotation branches use lightweight transposed-convolutional upsampling to produce logits

$$\mathbf{Z}_{\mathrm{tr}}, \mathbf{Z}_{\mathrm{rot}} \in \mathbb{R}^{k \times 3 \times R}. \tag{2}$$

The transposed-convolutional structure provides the key inductive bias. Because adjacent bins correspond to nearby continuous actions, convolutional weight sharing encourages neighboring bins to receive correlated evidence rather than be treated as unrelated classes. This gives the decoder a native mechanism for modeling spatial uncertainty characterized by local correlations among neighboring action bins.

The logits are normalized into categorical action probabilities

$$\mathbf{P}_{\mathrm{tr}} = \mathrm{softmax}(\mathbf{Z}_{\mathrm{tr}}), \quad \mathbf{P}_{\mathrm{rot}} = \mathrm{softmax}(\mathbf{Z}_{\mathrm{rot}}), \tag{3}$$

where the softmax is applied along the bin dimension. A continuous action value is obtained as the expectation over bin centers,

$$a_{n,j} = \sum_{r=1}^{R} \mathbf{P}_{n,j,r} c_r, \tag{4}$$

for chunk index n and action axis j, where $\mathbf{P}$ denotes the corresponding translation or rotation probability tensor. Because this expectation is differentiable, L1 action supervision can optimize the decoder end to end. A lightly weighted auxiliary distribution loss further regularizes the predicted probabilities. For the gripper dimension, we use a small MLP rather than a transposed-convolutional distribution head, since binary or near-binary gripper commands have much weaker spatial uncertainty than translation and rotation.

3.2 SOFT-DTW ALIGNMENT

Robot demonstrations can be semantically consistent while differing in local execution speed. For example, one demonstration may approach an object slowly and then close the gripper quickly, while another may execute the same behavior at a more uniform speed. Framewise L1 supervision penalizes such local timing shifts even when the action chunks describe the same underlying behavior. Inspired by dynamic time warping in speech recognition (Sakoe & Chiba, 1978), we use Soft-DTW (Cuturi & Blondel, 2017) as a differentiable alignment loss between predicted and demonstrated action chunks.

Let $\mathcal{A}_t = \{a_i\}_{i=1}^k$ and $\mathcal{A}_t^* = \{a_j^*\}_{j=1}^k$ denote the predicted and demonstrated action chunks. Excluding the final gripper dimension, we form cumulative trajectories $\mathcal{X}_t = \{x_i\}_{i=1}^k$ and $\mathcal{X}_t^* = \{x_j^*\}_{j=1}^k$, where $x_i = \sum_{\ell=1}^{i} a_{\ell,1:d_a-1}$ and $x_j^* = \sum_{\ell=1}^{j} a_{\ell,1:d_a-1}^*$. We define the pairwise trajectory cost as

$$C_{ij} = \frac{1}{d_a - 1} \|x_i - x_j^*\|_2^2, \tag{5}$$

which measures the mean squared discrepancy between two candidate aligned steps. Soft-DTW replaces the hard minimum in DTW with a differentiable soft minimum. With $D_{0,0} = 0$ and $D_{i,0} = D_{0,j} = +\infty$, we compute

$$\begin{aligned} D_{i,j} &= C_{ij} + \mathrm{softmin}_\gamma(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1}), \\ \mathrm{softmin}_\gamma(z_1, z_2, z_3) &= -\gamma \log \sum_{m=1}^{3} \exp(-z_m/\gamma). \end{aligned} \tag{6}$$

The temperature γ controls the alignment smoothness, with smaller values approaching standard DTW. We denote the resulting cost by $D_\gamma(\mathcal{X}_t, \mathcal{X}_t^*) = D_{k,k}$.

To remove the self-similarity bias of the raw alignment score, we use the Soft-DTW divergence

$$\mathcal{L}_{\mathrm{div}} = D_\gamma(\mathcal{X}_t, \mathcal{X}_t^*) - \frac{1}{2} D_\gamma(\mathcal{X}_t, \mathcal{X}_t) - \frac{1}{2} D_\gamma(\mathcal{X}_t^*, \mathcal{X}_t^*). \tag{7}$$

This objective encourages the predicted cumulative trajectory to follow a temporally flexible monotonic alignment path to the demonstration. As a result, the policy can learn from semantically consistent action chunks even when their local timing differs.

3.3 SUPERVISION

For spatial uncertainty, we primarily supervise the continuously decoded actions with

$$\mathcal{L}_{\mathrm{L1}} = \frac{1}{k d_a} \sum_{n=1}^{k} \sum_{j=1}^{d_a} \left| a_{n,j} - a_{n,j}^* \right|, \tag{8}$$

where a and a^* are the predicted and demonstrated actions. Differentiable expectation-based decoding allows this loss to optimize the probabilistic decoder end to end. Assuming the gripper is the last action dimension, we further impose a weak KL constraint on the remaining $d_a - 1$ translation and rotation dimensions:

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{k(d_a-1)} \sum_{n=1}^{k} \sum_{j=1}^{d_a-1} D_{\mathrm{KL}}(\mathbf{q}_{n,j} \,\|\, \mathbf{P}_{n,j}) \,. \tag{9}$$

Here, $\mathbf{P}$ is the predicted distribution and $\mathbf{q}$ is a discretized Gaussian centered at the demonstrated action, with standard deviation σ as a bandwidth hyperparameter. The divergence is

$$D_{\mathrm{KL}}(\mathbf{q} \,\|\, \mathbf{P}) = \sum_{r=1}^{R} q_r \log \frac{q_r}{P_r}. \tag{10}$$

We set $\lambda_{\mathrm{KL}} = 0.05$ empirically, so KL serves only as a stabilizing regularizer while L1 remains the primary spatial supervision.

For temporal uncertainty, we combine the Soft-DTW divergence $\mathcal{L}_{\mathrm{div}}$ with an endpoint loss

$$\mathcal{L}_{\mathrm{end}} = \|x_k - x_k^*\|_1 \,, \tag{11}$$

where x_k and x_k^* are the final cumulative trajectory points of the predicted and demonstrated chunks. Soft-DTW tolerates local timing uncertainty, while the endpoint term preserves final task progress.

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{L1}} + \lambda_{\mathrm{KL}} \mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{DTW}} \left(\mathcal{L}_{\mathrm{div}} + \mathcal{L}_{\mathrm{end}} \right). \tag{12}$$

We set $\lambda_{\mathrm{DTW}} = 0.5$ empirically. We optimize all loss terms jointly throughout training and anneal the Soft-DTW smoothing coefficient γ in Eq. 6 from a larger initial value to its target value, transitioning from smooth alignment to more precise temporal matching.

4 Experiments

We evaluate SURE on the LIBERO and RoboTwin 2.0 simulation benchmarks and on six real-world manipulation tasks using demonstrations collected for this study. Implementation details for these evaluations are provided in Appendix A.

4.1 Simulation Evaluation

4.1.1 LIBERO

Evaluation protocol. We evaluate SURE on the four standard LIBERO task suites, namely LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. These suites collectively assess spatial reasoning, object-centric generalization, goal understanding, and long-horizon manipulation (Liu et al., 2023). The baseline results are collected from sources that use different statistical protocols. All results marked with † in Table 1 are averaged over three random seeds, with 500 evaluation trials per task suite (50 trials for each of the 10 tasks). Specifically, the results for Diffusion Policy, Octo, and OpenVLA are taken from the OpenVLA evaluation (Kim et al., 2025b). The GR00T-N1 and π_0 baselines are trained and reported by the authors of Hume (Song et al., 2025). OpenVLA-OFT and SURE are evaluated using a single seed.

Baseline methods. We compare SURE with a diverse set of representative policies spanning diffusion-based control, generalist robot policies, and VLA fine-tuning methods. For OpenVLA-OFT, we fine-tune the model using its publicly released implementation (Kim et al., 2025a) and report results after both 10k and 50k optimization steps. SURE uses the same OpenVLA-7B backbone and is trained for only 10k optimization steps.

Results. As shown in Table 1, SURE achieves the highest average success rate of 97.0% after only 10k optimization steps, surpassing OpenVLA-OFT by 9.9% under the same training budget and achieving gains across all four task suites. SURE also surpasses the 50k-step OpenVLA-OFT model by 2.7% in average success rate despite using only 20% optimization steps. Across individual suites, SURE obtains the best performance on all four task suites, surpassing the baselines on LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. These results indicate that explicitly modeling spatiotemporal uncertainty improves both final manipulation performance and training efficiency.

Table 1: Success rates (%) on the four LIBERO task suites. Training steps and batch sizes (BS) are reported when available. SURE is trained for 10k optimization steps and achieves the highest average success rate. Bold values indicate the best result in each column. † denotes results averaged over three random seeds, all other results are obtained using a single evaluation seed.

Methods	Steps(BS)	Spatial	Object	Goal	Long	Avg.
Diffusion Policy† (Chi et al., 2023)	100-250k(64)	78.3	92.5	68.3	50.5	72.4
Octo† (Ghosh et al., 2024)	100-250k(64)	78.9	85.7	84.6	51.1	75.1
OpenVLA† (Kim et al., 2025b)	50-150k(64)	84.7	88.4	79.2	53.7	76.5
GR00T-N1† (NVIDIA et al., 2025)	/ (128)	94.4	97.6	93.0	90.6	93.9
π_0† (Black et al., 2025)	/ (128)	96.8	98.8	95.8	85.2	94.2
OpenVLA-OFT (Kim et al., 2025a)	10k(48)	94.4	92.6	92.6	68.8	87.1
OpenVLA-OFT	50k(48)	96.8	97.4	98.0	85.0	94.3
SURE	10k(48)	**99.0**	**99.2**	**98.8**	**91.0**	**97.0**

Table 2: Success rates (%) on five RoboTwin 2.0 tasks under the Easy evaluation setting. All methods are trained on the clean subset with a separate checkpoint for each task. Training steps and batch sizes (BS) are reported for each method. Bold values indicate the best result in each column.

Methods	Training steps(BS)	Beat Block Hammer	Click Alarm Clock	Lift Pot	Place Fan	Stack Bowls Two	Avg.
Diffusion Policy	20-75k(128)	42.0	61.0	39.0	3.0	61.0	41.2
OpenVLA-OFT	100k(8)	18.0	85.0	93.0	25.0	75.0	59.2
RDT (Liu et al., 2025)	10k(64)	**77.0**	61.0	72.0	12.0	76.0	59.6
π_0	30k(32)	43.0	63.0	84.0	20.0	**91.0**	60.2
SURE	20k(16)	55.0	**100.0**	**98.0**	**26.0**	82.0	**72.2**

4.1.2 RoboTwin 2.0

Evaluation protocol. RoboTwin 2.0 is a large-scale bimanual manipulation benchmark containing 50 tasks and supporting multiple dual-arm embodiments (Chen et al., 2026). We evaluate on the five tasks listed in Table 2: Beat Block Hammer, Click Alarm Clock, Lift Pot, Place Fan, and Stack Bowls Two. All methods are trained on the clean subset, which contains 50 clean expert demonstrations for each task. Following the single-task setting, we fine-tune or train a separate checkpoint for every task. We report the success rate over 100 evaluation rollouts per task under the Easy setting, which uses clean, non-randomized environments.

Baseline methods. We compare SURE with Diffusion Policy, RDT(Liu et al., 2025), π_0, and OpenVLA-OFT. The results for Diffusion Policy, RDT, and π_0 are taken from the RoboTwin 2.0 paper (Chen et al., 2026). The OpenVLA-OFT results are taken from JEPA-VLA (Miao et al., 2026). These baselines cover diffusion-based visuomotor control, pretrained vision-language-action policies, and efficient fine-tuning of an OpenVLA backbone.

Results. As shown in Table 2, SURE achieves the highest average success rate of 72.2%, outperforming the strongest baseline, π_0, by 12.0%. SURE obtains the best performance on three of the five tasks, reaching 100.0% on Click Alarm Clock, 98.0% on Lift Pot, and 26.0% on Place Fan. It also achieves the second-best result on Beat Block Hammer and Stack Bowls Two. Compared with OpenVLA-OFT, SURE improves the average success rate by 13.0% while using only 20k rather than 100k optimization steps. These results demonstrate that SURE improves both manipulation performance and training efficiency on bimanual tasks.

4.2 Real-World Evaluation

Evaluation protocol. We evaluate SURE on six real-world manipulation tasks: Water Bottle, AAA Battery, Screwdriver, Toy Duck, Stack Bowls Two, and Pour Water. For each task, we collect 50 demonstration episodes through multiple authors acting as operators. Differences in operator motion and execution speed introduce natural spatial and temporal uncertainty into the demonstrations.

Table 3: Success rates (%) on six real-world manipulation tasks. Each method is evaluated over 20 trials per task. OpenVLA-OFT and SURE are trained for 50k and 20k optimization steps, respectively. Bold values indicate the best result in each column.

Methods	Water Bottle	AAA Battery	Screw-driver	Toy Duck	Stack Bowls Two	Pour Water	Avg.
OpenVLA-OFT	40	35	55	45	20	**55**	41.7
SURE	**55**	**60**	**70**	**95**	**25**	40	**57.5**

Each policy is evaluated over 20 trials per task, and we report the corresponding success rate. Detailed descriptions of the task setups and execution procedures are provided in the AppendixC.

Baseline method. We compare SURE with OpenVLA-OFT using the same mixed-task real-world demonstrations. Both methods are jointly trained on demonstrations from all six tasks, and a single checkpoint is evaluated across all tasks. OpenVLA-OFT is trained for 50k optimization steps, whereas SURE is trained for only 20k steps.

Results. As shown in Table 3, SURE achieves an average success rate of 57.5%, outperforming OpenVLA-OFT by 15.8% while using 40% optimization steps. SURE performs better on five of the six tasks, with the largest improvement on Toy Duck, where the success rate increases from 45% to 95%. These results demonstrate that SURE learns effectively from real-world demonstrations with substantial spatial and temporal uncertainty.

4.3 Inference Efficiency Evaluation

Evaluation protocol. We evaluate the inference efficiency of the action head using models trained on LIBERO-Spatial, with the action chunk length fixed to $k = 8$. To isolate the computational cost of action decoding, we measure only the action head or decoder and exclude the VLA backbone. Because $\pi_{0.5}$ (Physical Intelligence et al., 2025) is implemented in JAX, we additionally implement our action head in JAX to provide a framework-matched latency comparison. For each method, we report the number of action-head parameters, FLOPs, and mean chunk latency. Latency is averaged over 100 action-decoding runs following 20 warm-up runs.

Results. As shown in Table 4, SURE contains only 16.93M action-head parameters and requires 10.96G FLOPs, corresponding to 3.9% of the parameters and 13.8% of the FLOPs of $\pi_{0.5}$. The standard SURE implementation generates an action chunk in 1.57 ms, providing a 12.7× speedup over $\pi_{0.5}$. Under the framework-matched JAX comparison, SURE reduces the latency to 0.69 ms, corresponding to a 28.8× speedup over $\pi_{0.5}$. Although the deterministic OFT head remains faster and requires fewer FLOPs, SURE uses only 11.2% as many parameters while retaining an uncertainty-aware action representation. These results show that SURE provides efficient uncertainty-aware action decoding without the high inference cost of iterative generative action heads.

Table 4: Comparison of action head complexity and inference efficiency. Stochastic indicates whether the action head explicitly models the action distribution. Params denotes the number of parameters in the action head. Latency denotes the mean time required by the action head to decode one action chunk, excluding the VLA backbone. SURE* denotes our JAX variant used for a framework-matched comparison with $\pi_{0.5}$.

Stochastic	Methods	Params	FLOPs	Latency
No	OFT	151.12M	2.42G	0.37ms
Yes	$\pi_{0.5}$ (JAX)	430.10M	79.13G	19.88ms
	SURE	16.93M	10.96G	1.57ms
	SURE* (JAX)	16.93M	10.96G	0.69ms

4.4 Data Efficiency Evaluation

Evaluation protocol. To evaluate data efficiency independently of the optimization budget, we train SURE and OpenVLA-OFT on LIBERO-Spatial using 10%, 25%, 50%, and 100% of the available

training demonstrations. We fix the training budget at 10k optimization steps for all data ratios and keep the remaining training settings unchanged. OpenVLA-OFT serves as the baseline because it uses the same OpenVLA-7B backbone and provides a strong fine-tuning recipe for continuous action prediction (Kim et al., 2025a). All results in this evaluation are obtained using a single seed.

Results. As shown in Figure 3, SURE consistently outperforms OpenVLA-OFT across all training data ratios. With 10%, 25%, 50%, and 100% of the demonstrations, SURE improves the success rate by 21.4, 4.2, 4.2, and 4.6%, respectively. Notably, SURE reaches 95.0% success using only 25% of the training data, exceeding the 94.4% achieved by OpenVLA-OFT with the full dataset. Even with only 10% of the demonstrations, SURE achieves 93.8%, remaining within 0.6% of the full-data OpenVLA-OFT result. These results show that explicitly modeling spatial and temporal uncertainty allows SURE to learn effective manipulation policies from substantially fewer demonstrations.

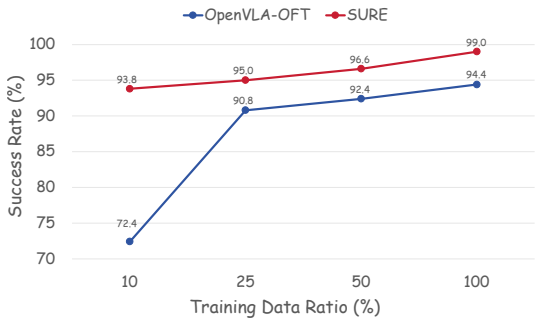


Figure 3: Success rates (%) on LIBERO-Spatial using different fractions of the available training demonstrations. Both methods are trained for 10k optimization steps.

4.5 Ablations

We conduct subtractive ablations on LIBERO-Spatial to evaluate the probabilistic action decoder and Soft-DTW alignment supervision. All variants are trained on LIBERO-Spatial for 10k optimization steps using the same settings.

As shown in Table 5, removing either component reduces the success rate. Removing Soft-DTW decreases performance from 99.0% to 97.4%, while removing the probabilistic action decoder yields 96.8%. Removing both components further reduces the success rate to 94.4%, confirming that spatial and temporal uncertainty modeling are complementary.

Table 5: Subtractive ablation on LIBERO-Spatial. Prob denotes the probabilistic action decoder, and DTW denotes Soft-DTW alignment supervision. All variants are trained for 10k steps.

Variant	Steps	Prob	DTW	Success Rate
SURE	10k	✓	✓	**99.0**
w/o DTW	10k	✓	–	97.4
w/o Prob	10k	–	✓	96.8
w/o Both	10k	–	–	94.4

5 Conclusion

We presented SURE, a training- and data-efficient VLA framework that explicitly models the spatial and temporal uncertainty inherent in robot demonstrations. SURE combines a lightweight probabilistic action decoder for spatial uncertainty with Soft-DTW alignment supervision that accommodates differences in execution timing. The decoder further supports fast parallel action prediction without iterative denoising. Experiments on LIBERO, RoboTwin 2.0, and six real-world tasks demonstrate improved task performance and data efficiency under reduced training budgets, together with efficient action decoding. Future work may extend SURE to larger cross-embodiment datasets and more diverse manipulation settings.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, manuscript organization, LaTeX formatting, and summarizing experiment records provided by the authors for presentation in the manuscript. We did not use generative AI tools to design the methodology, write or modify experimental code, run experiments, or generate experimental data or results. Other disclosure tasks are not applicable to this work. We reviewed all work produced with AI assistance. The authors checked text produced with AI assistance against the manuscript plan and original experiment records, verified citation metadata against primary sources, and compiled and visually inspected the affected pages. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work evaluates robot manipulation policies on the LIBERO and RoboTwin 2.0 simulation benchmarks and on six tabletop tasks using LeRobot SO-101 arms. The real-world demonstrations were collected by the authors as operators. The tasks involve robot interaction with objects rather than interaction with people. Physical deployment nevertheless carries risks from unintended arm motion, collisions, and spills during the pouring task. The reported task success rates do not measure operational safety or justify unsupervised use around people. Deployment beyond the evaluated settings would require task-specific safety assessment, appropriate physical safeguards, and human oversight.

REPRODUCIBILITY STATEMENT

The Methods section defines the probabilistic action decoder, Soft-DTW alignment, and training objective. Section 4 specifies the LIBERO, RoboTwin 2.0, and real-world evaluation protocols, including task selection, training budgets, evaluation trials, and the provenance of baseline results. It also describes the data-efficiency, ablation, and action-head latency evaluations. Appendix A reports the backbone adaptation, optimizer and loss settings, action chunk lengths, training schedules, hardware, and the RoboTwin-specific decoder. Appendix C describes the robot setup, demonstration collection, six task instructions, and representative rollouts.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://www.roboticsproceedings.org/rss19/p025.html.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable

data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Proceedings of International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62192.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In *Proceedings of International Conference on Machine Learning*, pp. 894–903, 2017. URL https://proceedings.mlr.press/v70/cuturi17a.html.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3D object manipulation. In *Proceedings of Conference on Robot Learning*, pp. 694–710, 2023. URL https://proceedings.mlr.press/v229/goyal23a.html.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *Proceedings of International Conference on Learning Representations*, pp. 6121–6138, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0ae94013da7cd459402fd77874e09ee3-Abstract-Conference.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603.

Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-Fine Q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13739–13748, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/James_Coarse-To-Fine_Q-Attention_Efficient_Learning_for_Visual_Robotic_Manipulation_via_Discretisation_CVPR_2022_paper.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of Conference on Robot Learning*, pp. 2679–2713, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng YAN, Maoqing Yao, and Guanghui Ren. Genie Envisioner: A unified world foundation platform for robotic manipulation. In *Proceedings of International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8e773e9e463e97b930c47621b6777159-Abstract-Conference.html.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information*

Processing Systems, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *Proceedings of International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/49f80e4d2471ad4f2edf4f5f1ab62339-Abstract-Conference.html.

Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. JEPA-VLA: Video predictive embedding is needed for VLA models. *CoRR*, 2026. URL https://arxiv.org/abs/2602.11832.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Jim Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *CoRR*, 2025. URL https://arxiv.org/abs/2503.14734.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari,

Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *Proceedings of International Conference on Robotics & Automation*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://ieeexplore.ieee.org/document/10611477.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

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

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/TASSP.1978.1163055. URL https://doi.org/10.1109/TASSP.1978.1163055.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-Actor: A multi-task transformer for robotic manipulation. In *Proceedings of Conference on Robot Learning*, pp. 785–799, 2023. URL https://proceedings.mlr.press/v205/shridhar23a.html.

Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, Dong Wang, and Xuelong Li. Hume: Introducing system-2 thinking in visual-language-action model. *CoRR*, 2025. URL https://arxiv.org/abs/2505.21432.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. TinyVLA: Toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025. doi: 10.1109/LRA.2025.3544909. URL https://ieeexplore.ieee.org/document/10900471/.

Pei Yang, Hai Ci, Yanzhe Chen, Qi Lv, Han Cai, and Mike Zheng Shou. ActionMap: Robot policy learning via voxel action heatmap. *CoRR*, 2026. URL https://arxiv.org/abs/2606.06904.

Chia-Han Yeh, Tse-Sheng Nan, Risto Vuorio, Wei Hung, Hung-Yen Wu, Shao-Hua Sun, and Ping-Chun Hsieh. Action-constrained imitation learning. In *Proceedings of International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/yeh25a.html.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. τ_0-WM: A unified video-action world model for robotic manipulation. *CoRR*, 2026. URL https://arxiv.org/abs/2606.01027.

A Implementation Details

Across all experiments, we adapt OpenVLA-7B using LoRA with rank 32 and $\alpha = 16$, and optimize the model with AdamW using a weight decay of 0.01. We use an action chunk length of $k = 8$ for LIBERO and the real-world experiments while $k = 25$ for RoboTwin 2.0. The loss weights are $\lambda_{\mathrm{KL}} = 0.05$ and $\lambda_{\mathrm{DTW}} = 0.5$, and the Gaussian target distribution uses $\sigma = 1.5$ bins.

LIBERO. We train one checkpoint for each task suite for 10k optimization steps on 4 NVIDIA H20 GPUs. We use a per-device batch size of 12, a learning rate of 5×10^{-4}, and an action chunk length of $k = 8$. The Soft-DTW smoothing coefficient γ is initialized to 0.1 and annealed to 0.005 over the first 5k optimization steps.

RoboTwin 2.0. Consistent with the baseline protocols, we train a separate checkpoint for each task. Each model is trained for 20k optimization steps on a single NVIDIA B300 GPU with a batch size of 16 and an action chunk length of $k = 25$. The learning rate is initialized to 5×10^{-4} and annealed to 5×10^{-5} starting at step 10k. We initialize γ to 0.1 and anneal it to 0.001 over the first 10k optimization steps.

LIBERO represents an action using end-effector translation, rotation, and a gripper command, whereas RoboTwin controls two arms using absolute joint angles. We therefore replace the translation–rotation decomposition in the probabilistic action decoder with a single 12-axis joint decoder, which predicts all 12 joint commands (six per arm) at each action step. Each joint is represented by a 256-bin distribution and decoded by expectation. A separate MLP branch predicts the two gripper commands, after which the output is rearranged to match RoboTwin's action convention, $[\mathbf{q}^L_{1:6}, g^L, \mathbf{q}^R_{1:6}, g^R]$. The stabilizing KL constraint is correspondingly applied to all 12 joint distributions. For the temporal objective, we first regroup each action as $[\mathbf{q}^L_{1:6}, \mathbf{q}^R_{1:6}, g^L, g^R]$ and apply Soft-DTW divergence to the predicted and ground-truth joint-action chunks.

Real-world experiments. We conduct real-world experiments using LeRobot SO-101 robotic arms. We train for 20k optimization steps on 2 NVIDIA A100 GPUs with a per-device batch size of 8, a learning rate of 5×10^{-4}, and an action chunk length of $k = 8$. Following the same Soft-DTW schedule as RoboTwin 2.0, we anneal γ from 0.1 to 0.001 over the first 10k optimization steps.

B Action Distributions Visualizations

B.1 LIBERO

Figure 4 visualizes representative SURE action distributions on the four LIBERO task suites. In each column, the top image shows the robot observation and the bottom heatmaps show the predicted distributions for translation, rotation, and gripper actions. Diamonds mark the actions obtained by expectation-based decoding.

B.2 RoboTwin 2.0

Figure 5 visualizes representative SURE joint-action distributions on four RoboTwin 2.0 tasks. In each column, the top image shows the bimanual robot observation, and the heatmaps below show the predicted action distributions for the left and right arm joints and grippers.

C Real-World Experiments

C.1 Setup Overview

We conduct real-world experiments with LeRobot SO-101 robotic arms. As shown in Figure 6, the setup provides a front camera and a wrist-mounted camera, which together capture the workspace and fine-grained manipulation details. We collect 50 demonstration episodes for each of the six tasks with multiple operators. The resulting variation in motion trajectories and execution speed provides natural spatial and temporal uncertainty. Both SURE and OpenVLA-OFT are jointly trained on the mixed-task demonstrations and evaluated using a single checkpoint. Each policy is evaluated over 20 trials per task, and we report the corresponding success rate.

C.2 INSTRUCTIONS

1. **Water Bottle:** Pick up the bottle and place it on the black plate.
2. **AAA Battery:** Pick up the AAA battery and place it in the white basket.
3. **Screwdriver:** Pick up the screwdriver and place it in the white basket.
4. **Toy Duck:** Pick up the toy duck and place it in the white basket.
5. **Stack Bowls Two:** Stack the red bowl onto the black bowl.
6. **Pour Water:** Pour the water from the water bottle into the cup.

C.3 VISUALIZATIONS

Figure 7 presents representative SURE rollouts for all six real-world manipulation tasks, illustrating the temporal progression of each task from the initial scene to the final state.

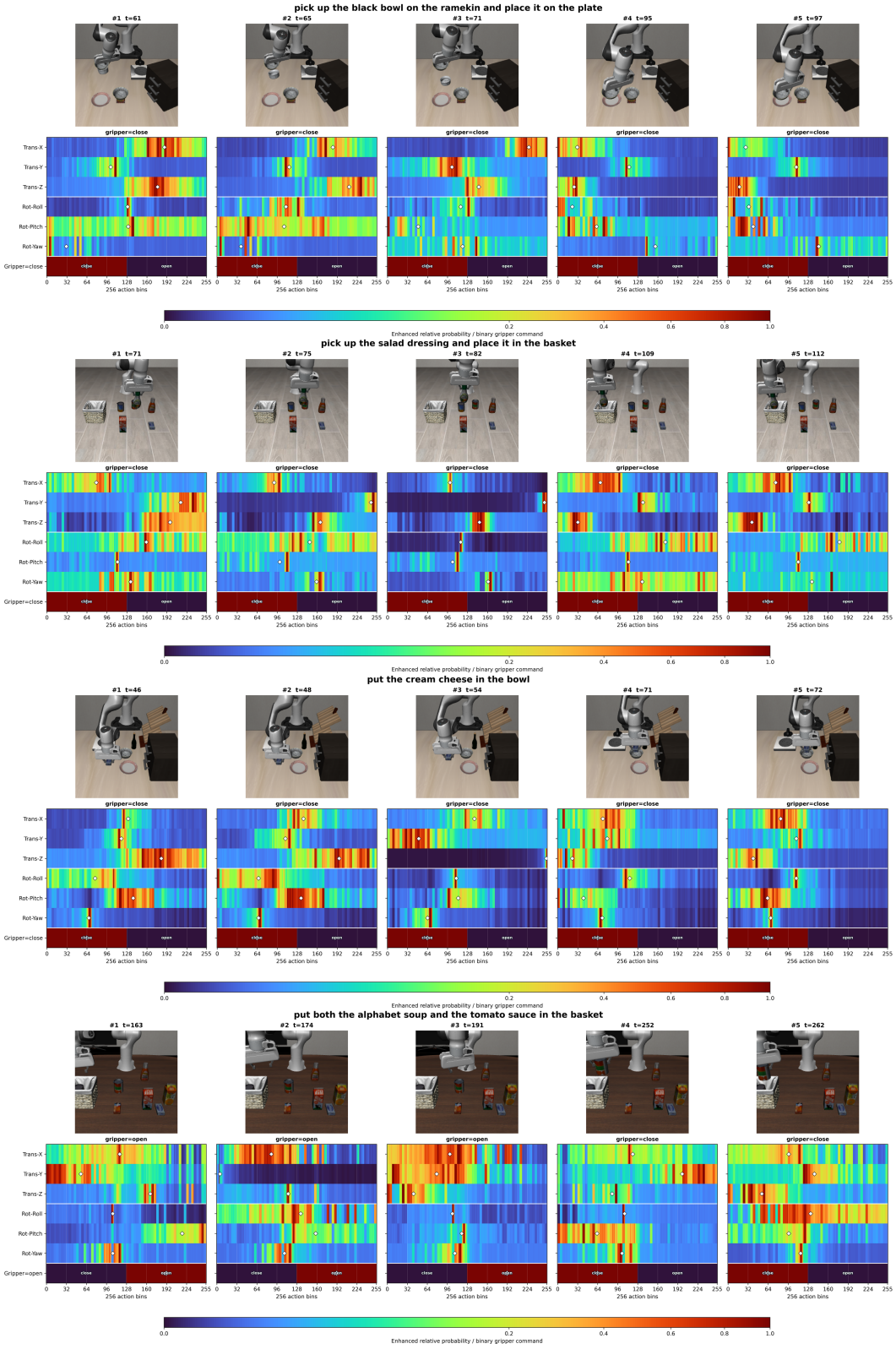


Figure 4: Representative SURE action-distribution visualizations on LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, from top to bottom.

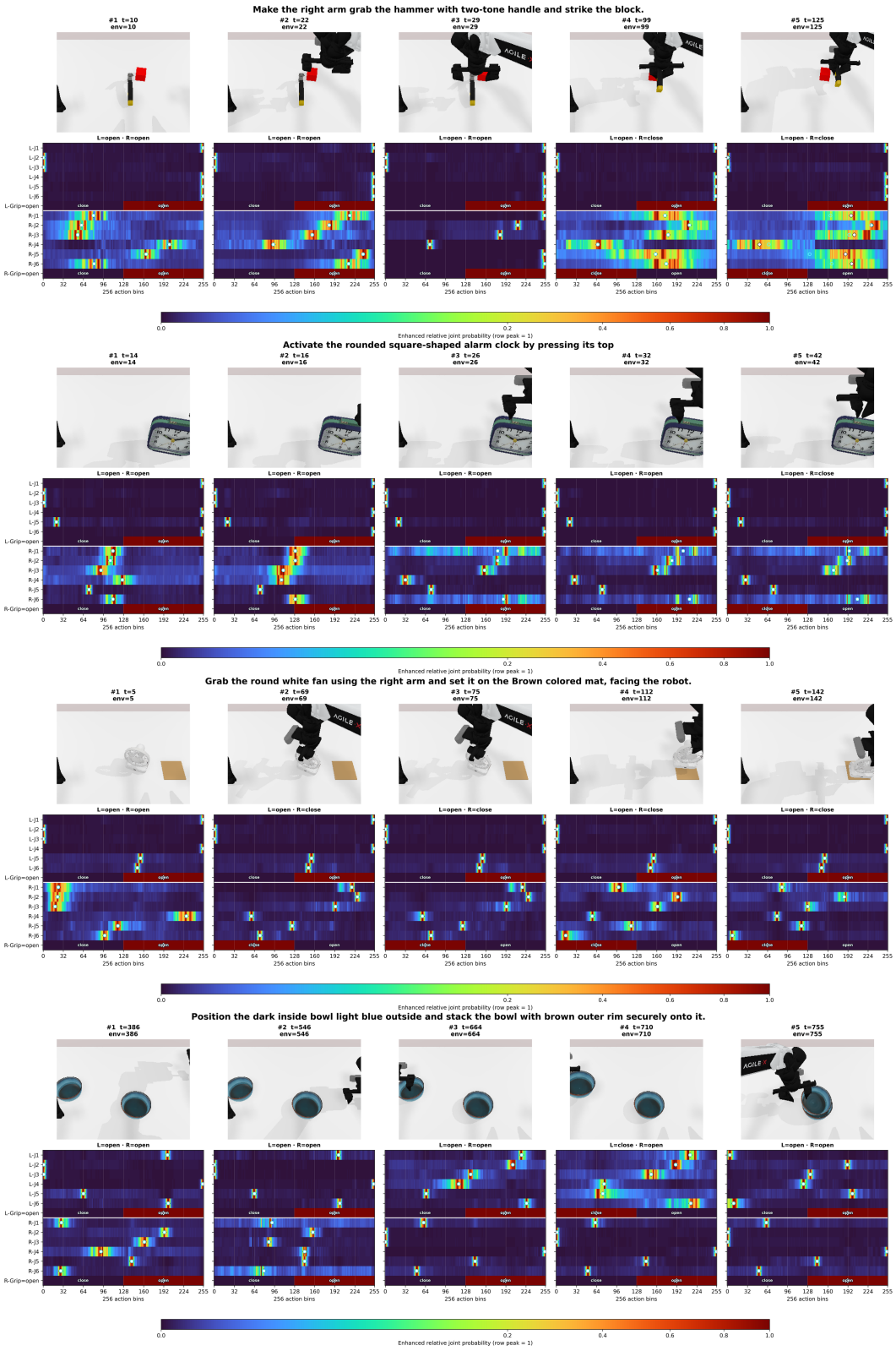


Figure 5: Representative SURE joint-action distribution visualizations on RoboTwin 2.0: Beat Block Hammer, Click Alarm Clock, Place Fan, and Stack Bowls Two, from top to bottom.

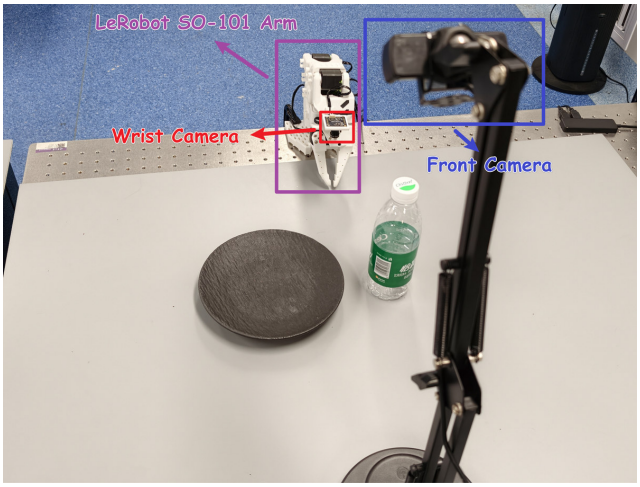


Figure 6: Real-world experimental setup with LeRobot SO-101 robotic arms. A front camera observes the workspace, while a wrist-mounted camera captures fine-grained manipulation details.

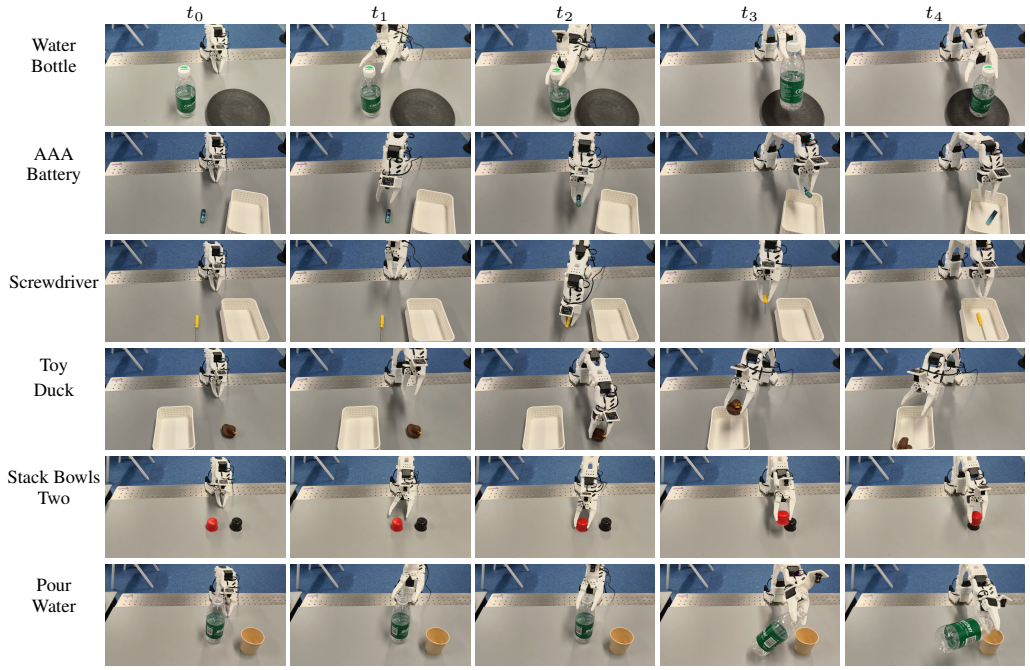


Figure 7: Real-world manipulation rollouts. Columns show the temporal progression from t_0 to t_4, while rows correspond to the six tasks. To provide clearer visualizations than the low-resolution front camera used by the policy, these frames are captured by an auxiliary camera placed alongside the front camera. Consequently, the robot arms are not centered in the images.