Trajectory-Level Continuous Action Representation for Robotic Manipulation

Tong Yang[1,3†], Jingkai Jia[2†], Yuecheng Xu[2†], Xueyao Chen[1], Chi Zhang[3], and Wenqiang Zhang[1,2‡]

[1] Shanghai Key Lab of Intelligent Information Processing, College of Computer Science and Artificial Intelligence, Fudan University, Shanghai, China
[2] College of Intelligent Robotics and Advanced Manufacturing, Fudan University, Shanghai, China
[3] TeleAI, China Telecom, Shanghai, China
tongyang23@m.fudan.edu.cn, wqzhang@fudan.edu.cn

Abstract. We propose CAT, a trajectory-level continuous action representation framework for robotic manipulation. Existing visuomotor systems often entangle action representation with control frequency or rely on fixed temporal parameterizations. This leads to representational redundancy at high sampling rates and limits the modeling of critical motion. CAT instead encodes action trajectories within a fixed real-time interval into a set of continuous latent tokens. To ensure temporal consistency across varying control frequencies, we further incorporate a frequency-aware positional encoding that establishs a shared temporal coordinate system. Trajectory-level regularization further stabilizes the latent representation. This approach prevents representation growth with timestep density and avoids reliance on predefined temporal parameterizations. Extensive system-level evaluations on LIBERO, MimicGen, and real-world long-horizon manipulation tasks demonstrate that CAT-based policies consistently outperform both competitive VQ-based and continuous visuomotor baselines under matched training settings. Across various model backbones and control frequencies, CAT consistently improves success rates. These results highlight the advantages of trajectory-level continuous action modeling for scalable robotic manipulation across varying control rates.

Keywords: Robotic Manipulation · Continuous Action Representation · Trajectory-Level Modeling

1 Introduction

Learning visuomotor policies requires modeling continuous action trajectories whose influence on task outcomes is often unevenly distributed over time. In many manipulation tasks, only certain trajectory segments substantially affect task success. These include contact transitions, switches between free motion and

† Equal contribution. ‡ Corresponding author.

interaction, and brief corrective adjustments. Other portions of the trajectory often consist of relatively smooth and repetitive motion. This temporal imbalance poses a challenge for policy learning. When modeling capacity is spread across the entire trajectory, temporally redundant regions can dominate the learning signal and dilute gradients associated with shorter but more consequential segments [23]. As a result, action representations that do not account for this imbalance may struggle to preserve information carried by localized but important action variations.

Control frequency further amplifies this limitation in action representation. Under timestep-level discretization [8, 12, 26], representational structure is directly tied to the temporal grid. As sampling frequency increases, trajectory length grows proportionally, expanding the timestep-level representation. However, task-relevant trajectory variation does not necessarily increase at the same rate, introducing temporal redundancy in the representation. To avoid this frequency-driven expansion, some approaches adopt predefined temporal reparameterizations [19, 23, 29]. These methods represent trajectories using a fixed set of parameters. While this prevents representation size from scaling with sampling frequency, the representational structure is still constrained by the chosen temporal basis.

To address these limitations, we introduce CAT, a trajectory-level continuous action representation framework. CAT encodes action trajectories over a fixed real-time interval into a compact set of continuous latent tokens. The number of tokens remains constant and does not scale with the temporal grid used for execution. CAT employs a frequency-aware positional encoding to normalize timestep indices by control frequency. This places trajectories sampled at different control rates in a shared temporal space. Frequency-dependent variations remain observable in this space and are reflected in the latent representation. This design allows CAT to learn trajectory-level latent representations without predefined temporal parameterizations. To ensure a stable latent structure, the representation is optimized using reconstruction and regularization objectives. These latent tokens serve as the action representation for policy learning and integrate seamlessly with a flow-matching policy learner.

We conduct extensive experiments in both simulated and real-world environments. On standard benchmarks including LIBERO [16] and MimicGen [20], we compare CAT-based policies with competitive visuomotor systems. These include VQ-based VLA/LLM-style approaches (e.g., FAST, VQ-VLA, and CARP) as well as continuous-control baselines. CAT achieves higher average success rates across tasks, demonstrating strong system-level performance. To evaluate behavior under varying control frequencies, we further conduct experiments on RoboTwin 2.0 [4], using it as a controlled multi-frequency evaluation platform. Across different sampling rates, CAT maintains consistent performance and generally outperforms baseline representations under matched training settings. We also validate CAT on real-world long-horizon manipulation tasks, where it achieves higher success rates than baseline systems, establishing a clear empirical advantage in extended-horizon manipulation.

In summary, our contributions are as follows:

- We introduce CAT, a trajectory-level continuous action representation framework. CAT encodes action trajectories using a fixed number of continuous latent tokens across control sampling rates. The representation is learned without relying on predefined temporal parameterizations.
- We develop a frequency-conditioned trajectory modeling architecture that incorporates control frequency as an input signal. This architecture establishes shared temporal coordinates across control rates while preserving control-rate variations in the representation.
- Through extensive system-level evaluations on both simulated and real-world robotic manipulation tasks, we show that CAT-based policies outperform competitive VQ-based and continuous-control baselines. The gains persist under varying control frequencies and in long-horizon real-robot settings.

2 Related Work

Vision-Language-Action Models. By leveraging the capabilities of pre-trained VLMs [10, 17, 21], Vision-Language-Action (VLA) models [1–3, 9, 11, 25] have emerged as a powerful paradigm for robotic manipulation, enabling the mapping of multimodal inputs into corresponding action outputs. Current approaches generally fall into two main paradigms: autoregressive models and diffusion-based models. Autoregressive models [2,3,11,25] predict actions step by step, typically representing actions as discrete tokens to leverage large language model architectures for sequential reasoning and policy generation. In contrast, diffusion-based approaches [1, 5, 9] directly model the denoising process in continuous control spaces, generating temporally coherent action trajectories. Despite their architectural differences, both paradigms model actions at the timestep level, with representational structures closely tied to temporal discretization.

Action Representation Learning. Prior VLAs commonly convert continuous action trajectories into structured sequence representations, which can broadly serve either discrete autoregressive prediction or continuous generative modeling. For discrete or autoregressive VLAs, action tokenizers provide a common way to convert continuous control signals into discrete symbolic sequences. Existing approaches either rely on predefined trajectory parameterizations, such as DCT coefficients [23] and B-spline control points [19], or learn vector-quantized codebooks over action sequences [7,8,12,18,22,26]. However, they bind representational structure to a specific temporal design—either via fixed basis functions, control-point allocations, or timestep-level discretization—causing the representation length to increase with finer sampling or to be constrained by the chosen parameterization. For continuous generative policies, such as diffusion- or flow-matching-based policies, avoid symbolic action tokens but typically predict timestep-aligned action chunks, keeping the representation coupled to temporal resolution. FreqPolicy [29] introduces continuous spectral-domain tokens, yet still relies on a predefined spectral basis. In contrast, CAT learns trajectory-level

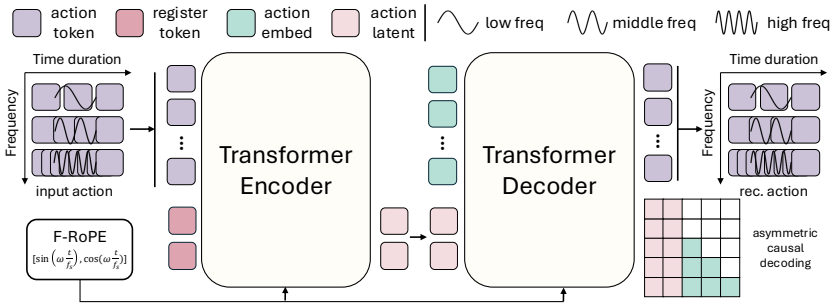


Fig. 1: Overview of CAT. CAT performs trajectory-level continuous action modeling over action sequences defined within a fixed real-time window. A frequency-aware transformer encoder with F-RoPE injection maps each action sequence to a fixed set of continuous latent variables. An asymmetric causal decoder reconstructs the action sequence in a frequency-conditioned manner, enabling temporally consistent modeling across varying control rates.

continuous latent tokens over a fixed real-time interval, with token dimensionality remaining constant across sampling rates. This enables scalable action modeling under varying control frequencies without timestep-level discretization or predefined temporal/spectral bases.

3 Method

3.1 Overview

CAT is designed to address structural dependencies introduced by predefined temporal designs in existing action representations. Prior approaches either allocate representational units according to the timestep grid or compress trajectories using fixed temporal parameterizations.

CAT removes this dependence through three key design principles. **Structural Decoupling** encodes trajectories over a fixed real-time interval into a fixed set of continuous latent tokens independent of timestep density. **Frequency Alignment** normalizes timesteps to align trajectories sampled at different control rates within a shared temporal space. **Latent Stability** combines reconstruction and contrastive regularization to maintain an expressive and discriminative latent space under a fixed token budget.

As illustrated in Fig. 1, CAT consists of a frequency-aware transformer encoder and decoder, both of which incorporate frequency-scaled timestep embeddings. The encoder compresses action trajectories into latent tokens, and the decoder reconstructs continuous trajectories. The learned latent tokens can be integrated into diffusion-based VLA frameworks as compact action representations for downstream generation.

3.2 Trajectory-level Continuous Action Representation

Structural Decoupling: Existing action representations often derive their structure from a predefined temporal design. Under timestep-level discretization, representational units scale proportionally with the temporal grid. Alternatively, fixed temporal reparameterizations such as spectral bases or control-point schemes allocate parameters according to a predetermined basis. As a result, representational capacity becomes tied to the chosen temporal parameterization.

CAT removes this dependence by encoding trajectories within a fixed real-time window (T = 1 s) into a fixed number of continuous latent tokens. Given an action sequence $\{a_t\}_{t=1}^{N}$ and their associated frequency-scaled timestep indices $\{t/f_s\}_{t=1}^{N}$, CAT learns the mapping:

$$\{z_i\}_{i=1}^{K} = f_\theta(\{a_t\}_{t=1}^{N}, \{t/f_s\}_{t=1}^{N}), \quad K \ll N. \tag{1}$$

where $N = f_s T$ denotes the number of timesteps within the window under control frequency f_s, and token count K is fixed across sampling rates. The learnable register tokens $\{z_i\}_{i=1}^{K}$ attend to the full action sequence and aggregate trajectory-level information, ensuring that representational dimensionality does not scale with timestep density. The frequency-scaled timestep indices encode sampling-rate information and are used to construct frequency-aware positional embeddings.

To reconstruct actions, CAT employs a transformer decoder that takes as input the register tokens concatenated with a set of learnable action embeddings $\{e_t\}_{t=1}^{N}$. Each action embedding e_t corresponds to one timestep and is decoded into the reconstructed action $\hat{a}_t$. The overall decoding process can be expressed as:

$$\{\hat{a}_t\}_{t=1}^{N} = g_\phi(\{z_i\}_{i=1}^{K}, \{e_t\}_{t=1}^{N}, \{t/f_s\}_{t=1}^{N}) \tag{2}$$

where g_ϕ is the transformer decoder parameterized by ϕ. We design a decoder attention mask to regulate the information flow: all action embeddings can attend to the register tokens, while the action embeddings themselves follow causal attention along the temporal dimension. This asymmetric masking scheme enforces temporal causality during decoding, ensuring that action dependencies are modeled sequentially and improving the stability of generated trajectories.

However, structural decoupling alone does not resolve inconsistencies arising from different sampling densities. We therefore introduce an explicit frequency alignment mechanism.

Frequency Alignment: Under different control frequencies, the same timestep index corresponds to different physical times, making direct positional encoding inconsistent across control rates. To account for this discrepancy, CAT incorporates control frequency into positional encoding through a Frequency-aware Rotary Positional Embedding (F-RoPE). Each timestep index t is scaled to t/f_s, producing time coordinates with spacing $1/f_s$ and making the positional encoding frequency-aware. Formally, given a timestep t, its rotary embedding is defined as:

$$\text{F-RoPE}\left(\frac{t}{f_s}\right) = \left[\sin(\omega_i \tfrac{t}{f_s}),\ \cos(\omega_i \tfrac{t}{f_s})\right]_{i=1}^{d/2}, \tag{3}$$

where $\boldsymbol{\omega_i}$ denotes the i-th frequency coefficient, and d is the embedding dimension. These embeddings are then applied to the action query and action key vectors of the transformer layers in CAT. This embeds control-frequency information directly into the action representation, allowing trajectories sampled at different control rates to be distinguished.

Latent Stability and Training Objective: Encoding trajectories into a fixed number of latent tokens imposes a strong compression constraint: long action trajectories must be represented within a compact continuous latent space. Under this constraint, minimizing only reconstruction loss can cause distinct trajectories to occupy overlapping regions in latent space, reducing trajectory-level discriminability and weakening the stability of downstream generation.

To preserve discriminative structure within the fixed token budget, CAT combines reconstruction with trajectory-level contrastive regularization. Given an input action sequence $\{a_t\}_{t=1}^{N}$ and the decoder reconstruction $\{\hat{a}_t\}_{t=1}^{N}$, we use a reconstruction loss that penalizes discrepancies between the original and generated action trajectories:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{t=1}^{N} ||a_t - \hat{a}_t||_2^2. \tag{4}$$

However, reconstruction alone does not explicitly regulate the global structure of the latent space. As a result, semantically distinct trajectories may become insufficiently separated in the latent representation. To encourage discriminative trajectory-level embeddings under a fixed representational budget, we introduce a contrastive regularization term. For each trajectory, we first obtain its embedding $\bar{z}$ by concatenating all register tokens, $\bar{z} = [z_1||z_2||\cdots||z_K]$. The contrastive regularization is formulated as:

$$\mathcal{L}_{\text{reg}} = \frac{1}{B}\sum_{i=1}^{B} \log \sum_{\substack{j=1 \\ j\neq i}}^{B} \exp\left(-\frac{\mathcal{D}(\bar{z}_i, \bar{z}_j)}{\eta}\right) \tag{5}$$

where B denotes the mini-batch size, i indexes the anchor trajectory embedding, and j indexes the remaining samples within the same mini-batch. The function$\mathcal{D}(\cdot,\cdot)$ is a distance metric (e.g., cosine or Euclidean distance), and η is a temperature hyperparameter controlling the dispersion strength. This objective penalizes small inter-trajectory distances, ensuring that trajectories with different motion semantics occupy distinct regions in the latent space while maintaining intra-trajectory cohesion enforced by $\mathcal{L}_{\text{rec}}$.

The overall training objective is formulated as:

$$\mathcal{L}_{\text{CAT}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} \tag{6}$$

where λ_{reg} balances reconstruction accuracy and trajectory-level contrastive learning. Together, these two terms encourage CAT to learn a compact yet discriminative continuous action representation.

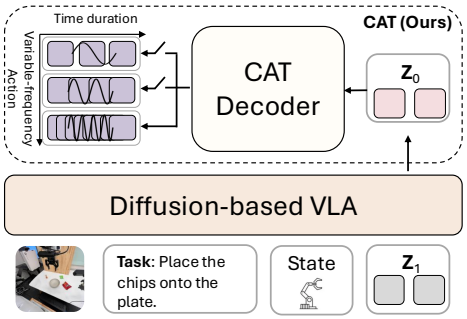


Figure 2. Integration of CAT into a diffusion-based VLA framework. CAT replaces the raw action space with a compact continuous latent space, decoupling representational complexity from control frequency. Control frequency is incorporated as a conditioning signal, allowing CAT-based policies to operate consistently under varying execution rates within diffusion-based VLA systems.

3.3 Integration into Diffusion-Based VLA

To demonstrate compatibility with continuous generative policies, we integrate CAT into a diffusion-based VLA framework. Diffusion-based VLAs generate temporally coherent action trajectories through iterative denoising, making them a natural setting to evaluate trajectory-level latent representations. CAT replaces the timestep-level action parameterization with a fixed-length continuous latent representation while preserving the original policy architecture.

The diffusion model, parameterized by ψ, operates on the latent trajectories $\mathbf{Z} = \{z_i\}_{i=1}^{K}$ and learns a continuous-time flow between noisy $\mathbf{Z}_1$ and clean latents $\mathbf{Z}_0$. At an intermediate time $\tau \in [0, 1]$, the latent state $\mathbf{Z}_\tau$ is defined as a linear interpolation between the clean and noisy endpoints: $\mathbf{Z}_\tau = (1-\tau)\mathbf{Z}_0 + \tau\mathbf{Z}_1$. The model predicts a time-dependent velocity field $v_\psi(\mathbf{Z}_\tau, \tau, \mathbf{c})$, where $\mathbf{c}$ represents task conditions such as language or scene context. Following the flow matching formulation, the training objective is defined as:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau,\, \mathbf{Z}_0,\, \mathbf{Z}_1}\left[\left\| v_\psi(\mathbf{Z}_\tau, \tau, \mathbf{c}) - (\mathbf{Z}_1 - \mathbf{Z}_0) \right\|_2^2\right] \tag{7}$$

During inference, as shown in Fig. 2, the diffusion model refines noisy latent trajectories into clean representations, which are decoded by CAT into continuous actions. The frequency-aware design of CAT enables consistent decoding under varying control rates while maintaining a fixed latent representation.

4 Experiments

We design our experiments to test structural hypotheses about trajectory-level action representation. Specifically, we evaluate whether modeling actions as trajectory-level continuous latent representations—decoupled from timestep discretization and fixed temporal parameterizations—leads to measurable improvements in policy learning, frequency robustness, and real-world deployment.

We formulate three hypotheses:

– **H1: Trajectory-Level Advantage.** Replacing timestep-level or discretization action representations with a trajectory-level continuous latent representation improves end-to-end policy performance under matched training settings.

Table 1: Comparison on LIBERO. Methods are grouped according to their action generation paradigm (autoregressive vs. diffusion/flow-based). Avg. denotes average success rate (%).

Method	Spatial	Object	Goal	Long	Avg.
Autoregressive-based VLA					
Octo [25]	78.9	85.7	84.6	51.1	75.1
OpenVLA [11]	84.7	88.4	79.2	53.7	75.9
SmolVLA-FAST	87.0	93.0	90.0	68.0	84.5
π_0-FAST [23]	96.4	96.8	88.6	60.2	85.0
SmolVLA-VQ-VLA	90.0	94.0	86.0	71.0	85.3
OmniSAT [19]	94.1	98.7	94.6	86.0	93.4
Diffusion/Flow-based VLA					
Diffusion Policy [5]	78.3	92.5	68.3	50.5	72.4
4D-VLA [28]	93.8	92.8	95.6	86.5	92.2
CogACT [13]	97.2	98.0	90.2	88.8	93.2
SmolVLA [24]	90.0	96.0	92.0	71.0	87.3
SmolVLA-CAT (Ours)	95.0	97.0	94.0	77.0	90.8
π_0 [1]	96.8	98.8	**95.8**	85.2	94.2
π_0-CAT (Ours)	**98.2**	**99.8**	95.6	**92.4**	**96.5**

– **H2: Frequency Robustness.** A trajectory-level representation with fixed latent budget remains stable under varying control frequencies, mitigating performance sensitivity to execution resolution.
– **H3: Long-Horizon Stability.** Trajectory-level continuous modeling improves temporal coherence and stability in long-horizon real-world manipulation tasks under identical training budgets.

We evaluate these hypotheses through controlled experiments in simulation and real-world settings, including comparisons against timestep-level continuous policies, discretization-dependent action representations, and fixed-basis trajectory parameterizations.

4.1 Simulation Experiments

We evaluate CAT in simulation on two widely used robotic manipulation benchmarks: LIBERO [16] and MimicGen [20]. These benchmarks cover diverse manipulation settings, including spatial tasks, goal-conditioned control, long-horizon sequential manipulation, and multi-task learning.

Evaluation on the LIBERO Benchmark

Setup. We evaluate CAT on the LIBERO benchmark [16], which contains four task suites: Spatial, Object, Goal, and Long.

For controlled comparisons, we construct four SmolVLA-based variants that share the same backbone architecture, policy network structure, dataset, preprocessing, and training schedule: (i) the original SmolVLA baseline with timestep-level continuous action prediction; (ii) SmolVLA-FAST and SmolVLA-VQ-VLA, which adopt discretization-dependent trajectory representations with autoregressive token prediction; and (iii) SmolVLA-CAT, which models each action chunk using continuous trajectory-level latent tokens. Across these variants, the network architecture remains identical; the only differences lie in the action representation and the corresponding prediction objective.

To evaluate cross-model generality, we further replace the action representation of the larger diffusion-based model π_0 [1] with CAT while keeping its backbone architecture, optimizer, and training protocol unchanged.

We also report representative autoregressive and diffusion-based VLA systems for context. These methods may differ in backbone scale and training data and are not intended as controlled comparisons.

Results. Table 1 reports the LIBERO results. Under the SmolVLA backbone, replacing timestep-level continuous action prediction with CAT improves the average success rate from 87.3% to 90.8% (+3.5). This controlled comparison highlights the benefit of shifting from timestep-level modeling to trajectory-level continuous latent representation, suggesting that allocating modeling capacity at the trajectory level leads to more effective policy learning.

Under the same backbone and identical policy architecture, SmolVLA-CAT consistently outperforms SmolVLA-FAST and SmolVLA-VQ-VLA. The consistent gains suggest that modeling actions as continuous trajectory-level latents provides a more effective inductive bias than discretization-dependent trajectory summaries—such as predefined-basis or timestep-level representations—within the same architecture.

Moreover, integrating CAT into the larger diffusion-based model π_0 further improves performance from 94.2% to 96.5%, indicating that the representational gain generalizes across model scales.

Overall, these results show that modeling actions as continuous trajectory-level latents provides a more effective inductive bias for visuomotor policy learning. This leads to consistent improvements across different architectures and model scales.

Evaluation on the MimicGen Benchmark

Setup. We evaluate CAT on MimicGen [20], a large-scale imitation learning benchmark built on Robomimic with diverse manipulation tasks and broad initial-state distributions. Following prior multi-task settings [8, 14, 27], we train a single policy jointly on 8 robosuite tasks.

We compare five representative methods: task conditioned diffusion (TCD) [14] and sparse diffusion policy (SDP) [27] as prior baselines, CARP [8] as an autoregressive multi-task policy with VQ-based action discretization, a flow-matching

Table 2: Comparison of success rates (%) on 8 manipulation tasks in MimicGen [20]. Bold numbers indicate the best performance.

Policy	Coffee	Hammer	Mug	Nut	Square	Stack	Stack 3	Threading	Avg.
TCD [14]	77	92	53	44	63	95	62	56	67.3
SDP [27]	82	**100**	62	54	82	96	80	70	78.3
CARP [8]	86	98	**74**	78	90	**100**	82	70	84.8
DP [5]	89	99	67	**84**	90	99	76	**82**	85.8
DP-CAT (Ours)	**91**	99	68	81	**91**	99	**85**	81	**86.9**

Diffusion Policy (DP) that replaces the original diffusion objective with flow matching [15], and DP-CAT, which substitutes timestep-level action prediction with trajectory-level continuous latent modeling.

For fair comparison, DP matches the parameter count of CARP. All models are trained under identical data splits and evaluation protocols. We report average success rates over 50 rollouts per task.

Results. Table 2 summarizes performance across 8 MimicGen tasks. The flow-matching Diffusion Policy (DP) achieves an average success rate of 85.8%, while integrating CAT improves performance to 86.9% under matched parameter counts (+1.1). This indicates that trajectory-level continuous latent modeling provides consistent gains over timestep-level prediction in a shared multi-task architecture.

The largest improvement appears on the Stack 3 task, where DP-CAT achieves 85% compared to 76% for DP (+9). Stack 3 involves sequential multi-stage stacking and longer effective control horizons. The observed gain suggests that trajectory-level representation remains effective in extended sequential settings.

Control-Frequency Evaluation on RoboTwin 2.0

Setup. To evaluate CAT under varying control frequencies, we repurpose RoboTwin 2.0 [4] as a controlled testbed for frequency analysis. Although RoboTwin is not originally designed as a control-frequency benchmark, its simulation framework enables systematic variation of control rates while keeping task configurations fixed. Experiments are conducted under the Easy split, as our focus is representation–frequency decoupling rather than scene generalization.

We collect manipulation trajectories at four control frequencies: 10Hz, 16.7Hz, 25Hz, and 50Hz. All trajectories are represented in the delta joint format to maintain a unified action space across frequencies. Using the aggregated multi-frequency dataset, we first train CAT as a shared action representation model. Based on the learned representation, we then train a separate SmolVLA-CAT policy for each control frequency, following the same backbone architecture, optimizer, and training protocol as the corresponding baseline SmolVLA model. Evaluation is conducted on a subset of 8 manipulation tasks from RoboTwin 2.0, with 100 rollouts per task. We report average success rates across tasks.

Table 3: Success rates (%) on 8 RoboTwin 2.0 manipulation tasks evaluated under different control frequencies. Results are averaged over 100 rollouts per task. Higher values between SmolVLA (Baseline) and SmolVLA-CAT (Ours) are highlighted in bold.

Task	10Hz		16.7Hz		25Hz		50Hz		Avg.(Task)	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
Adjust Bottle	**76**	68	77	**78**	69	**72**	78	**86**	75.0	**76.0**
Click Alarmclock	**54**	47	52	**54**	51	**58**	43	**50**	50.0	**52.3**
Grab Roller	17	**47**	46	**66**	49	**58**	56	**71**	42.0	**60.5**
Lift Pot	13	**25**	26	**28**	28	**29**	20	**27**	21.8	**27.3**
Place Burger Fries	**38**	37	24	**26**	**29**	**29**	26	**31**	29.3	**30.8**
Place Container Plate	43	**71**	**64**	57	50	**76**	**46**	40	50.8	**61.0**
Press Stapler	33	**52**	36	**52**	19	**50**	**50**	48	34.5	**50.5**
Rotate QRcode	8	**18**	**12**	8	8	**12**	1	**9**	7.3	**11.8**
Avg.(Frequency)	35.3	**45.6**	42.1	**46.1**	37.9	**48.1**	40.0	**45.3**	-	-

Results. Table 3 compares SmolVLA and SmolVLA-CAT under four control frequencies on RoboTwin 2.0. SmolVLA-CAT achieves higher average success rates than the baseline at all evaluated frequencies. Specifically, SmolVLA-CAT attains average success rates of 45.6%, 46.1%, 48.1%, and 45.3% at 10Hz, 16.7Hz, 25Hz, and 50Hz, respectively, compared to 35.3%, 42.1%, 37.9%, and 40.0% for SmolVLA. This corresponds to gains of +10.3, +4.0, +10.2, and +5.3 points.

Beyond the average improvement, CAT outperforms the baseline in 24 out of 32 task–frequency pairs, indicating that the gain is broadly consistent rather than driven by a small number of outlier tasks. Averaged across frequencies, the largest task-level improvements appear on *Grab Roller* (+18.5), *Press Stapler* (+16.0), *Place Container Plate* (+10.3), which involve short but critical interaction transitions. In contrast, improvements are smaller on smoother tasks such as *Adjust Bottle* (+1.0) and *Place Burger Fries* (+1.5). This pattern supports the motivation of CAT: trajectory-level representations are most beneficial when success depends on localized but important action variations.

Interestingly, the optimal control frequency varies across tasks. Some tasks, such as *Grab Roller* and *Adjust Bottle*, benefit from higher control rates, while others such as *Place Container Plate* achieve their best performance at lower frequencies. This observation suggests that the temporal resolution required for effective control depends on task dynamics and interaction complexity.

4.2 Real-World Long-Horizon Manipulation

Setup. We evaluate CAT on four long-horizon real-world manipulation tasks (see Fig. 3): **Stack-5** (stack four cubes sequentially on a red base cube to form a five-cube tower), **Drawer** (open the top drawer, place a tissue inside, and close it; then open the middle drawer and remove a sponge), **Rope** (thread a rope through two rings sequentially), and **Flower** (pick and place two roses into a vase, one at a time). These tasks require sequential completion of multiple

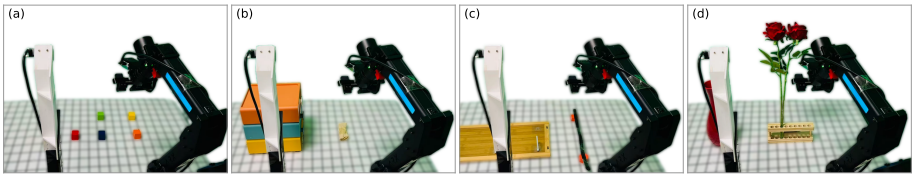


Fig. 3: Examples of initial states for 4 real-world tasks. Each panel shows the third-person camera, the robot arm with the wrist camera mounted, and the manipulated objects. Tasks: (a) Stack-5, (b) Drawer, (c) Flower, (d) Rope.

Table 4: Normalized task score (%) on 4 real-world long-horizon tasks. Baseline denotes SmolVLA, and Ours denotes SmolVLA integrated with CAT (SmolVLA-CAT). Bold numbers indicate the best performance.

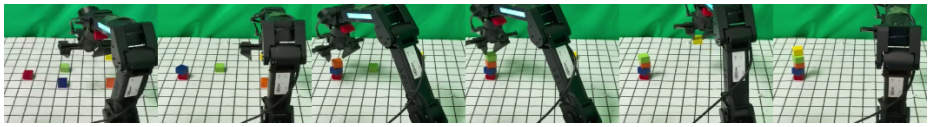
Models	Stack-5	Drawer	Rope	Flower	Avg.
Baseline	29	49	62	40	45.0
Ours	**43**	**61**	**70**	**68**	**60.5**

manipulation stages and precise control throughout execution. The experiments are conducted with a single ARX R5 arm equipped with one third-person camera and one wrist camera.

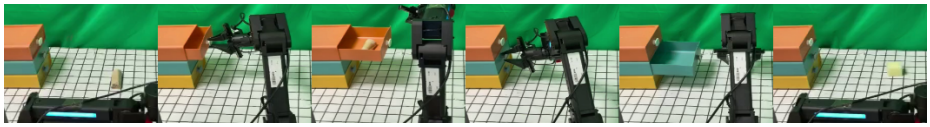
For each task, we collect 100 teleoperated demonstrations and train a task-specific policy. As the baseline, SmolVLA [24] is trained for 180k steps on each task. For our method, CAT is first pretrained on the Open X-Embodiment dataset [6] following the VQ-VLA [26] setup, then fine-tuned for 1k steps on the target task data, after which SmolVLA is trained using the CAT-based trajectory-level action representation for the same number of task-specific policy training steps as the baseline.

For evaluation, each task is decomposed into 4–8 sub-steps according to its execution horizon. Each completed sub-step contributes one point, and the total is normalized to a 0–100 scale, where 100 indicates that all sub-steps are completed. We report the average normalized task score over 10 evaluation episodes for each task.

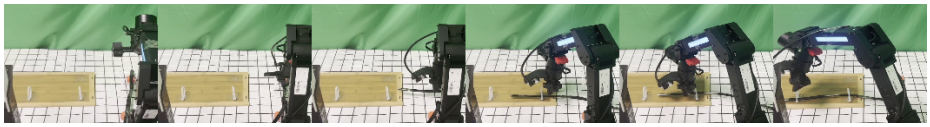
Results. Table 4 reports the normalized task scores on the four real-world tasks. CAT improves the average score from 45.0 to 60.5, yielding a gain of 15.5 points over the baseline. It outperforms the baseline on all four tasks, with gains of +14 on Stack-5, +12 on Drawer, +8 on Rope, and +28 on Flower. These results show that trajectory-level continuous latent modeling remains effective in real-world manipulation. All four tasks are long-horizon and require multiple sub-steps to be completed in sequence. The consistent improvements across stacking, articulated-object manipulation, and deformable-object manipulation suggest that CAT is particularly beneficial in real-world sequential manipulation scenarios where intermediate progress directly affects the final outcome.



(a) Stack-5: stack blocks in yellow-green-orange-blue-red order



(b) Drawer: open the top drawer, put the tissue into the top drawer and close drawer; open the middle drawer, take the sponge out and close the drawer.



(c) Rope: pick the rope, thread it into the first ring, drop it, pick it from the other side of the ring and thread it into the second ring.

(d) Flower: pick one flower and insert it into the vase; pick the flower left and insert it into the vase.

Fig. 4: Qualitative results on real-world tasks. Representative examples include tasks from Stack-5, Drawer, Rope, and Flower.

Representative qualitative examples of successful real-world executions are shown in Fig 4 for reference.

4.3 Ablation studies

To investigate the specific contributions of the key components in our proposed CAT, we conduct a series of ablation studies focusing on four major design factors: the positional encoding scheme, the number of registers K, the weight of the regularization loss λ_{reg}, and the decoder attention type. Unless otherwise specified, all ablations are performed on LIBERO under the same training and evaluation protocol as in the main experiments.

Positional Encoding. We compare frequency-aware rotary positional embeddings (F-RoPE) with learned positional embeddings on both LIBERO and RoboTwin 2.0. As shown in Fig. 5, F-RoPE gives slightly better performance on LIBERO and larger gains on RoboTwin 2.0, where evaluation spans multiple control frequencies. These results suggest that F-RoPE is a more effective positional encoding choice, particularly when the data and evaluation involve varying sampling rates.

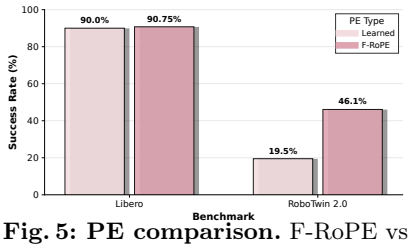


Fig. 5: PE comparison. F-RoPE vs learned PE.

Table 5: Ablation studies on LIBERO. S: Spatial, O: Object, G: Goal, L: Long-horizon. Performance of CAT under different design choices.

K	S	O	G	L	Avg
2	91	**98**	91	72	88.0
4	**95**	97	**94**	**77**	**90.8**
6	92	**98**	93	74	89.3
8	90	96	92	71	87.3

(a) Number of Registers

λ	S	O	G	L	Avg
0	91	**98**	93	75	89.3
0.1	**95**	97	**94**	**77**	**90.8**
0.2	92	97	92	76	89.3
0.3	90	95	91	76	88.0

(b) Reg. Loss Weight

Attn	S	O	G	L	Avg
Full	90	95	93	76	88.5
Causal	**95**	**97**	**94**	**77**	**90.8**

(c) Decoder Attention

Number of Registers. We vary the number of registers K to examine its influence on representation capacity. As shown in Tab. 5a, performance follows a clear non-monotonic trend. Increasing K from 2 to 4 improves the average success rate, while further increasing K leads to degradation. This indicates that simply enlarging the latent capacity does not monotonically improve performance; instead, there exists a moderate regime that balances expressiveness and optimization stability.

Regularization Loss Weight. We further analyze the effect of the regularization loss weight λ_{reg}, which controls the degree of latent separation in the learned action representations. As shown in Tab. 5b, setting $\lambda_{\text{reg}} = 0.1$ yields the highest average success rate. Removing the regularization term reduces average performance, while excessively large weights also degrade results. The optimal performance at $\lambda_{\text{reg}} = 0.1$ suggests that moderate regularization is beneficial without overly constraining the latent space.

Decoder Attention Type. We further explore how the attention mechanism in the decoder influences overall performance. As shown in Tab. 5c, causal attention consistently outperforms full attention across all task suites, improving the average success rate by 2.25 points. This result indicates that explicitly enforcing temporal directionality during decoding remains beneficial even when actions are represented at the trajectory level.

5 Conclusion

We introduce CAT, a trajectory-level continuous action representation framework for robotic manipulation. CAT represents action trajectories within a fixed real-time interval using continuous latent tokens. This formulation enables policies to operate robustly under varying control frequencies. Integrated with diffusion-based policy learning, CAT consistently improves policy performance across simulated benchmarks and real-robot manipulation tasks. These results highlight the effectiveness of trajectory-level continuous representations for visuomotor policy learning under varying control frequencies.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No.62576109), Scientific and Technological innovation action plan of Shanghai Science and Technology Committee (No.25511104402).

References

1. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: *pi*_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
2. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. arxiv. arXiv preprint arXiv:2307.15818 (2023)
3. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)
4. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., 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 (2025)
5. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research **44**(10-11), 1684–1704 (2025)
6. Collaboration, O.E., O'Neill, A., Rehman, A., Gupta, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models. arXiv preprint arXiv:2310.08864 **1**(2) (2023)
7. Dong, Z., Liu, Y., Zhang, S., Ye, B., Yuan, Y., Ni, F., Gong, J., Qiu, X., Zhao, H., Li, Y., et al.: Actioncodec: What makes for good action tokenizers. arXiv preprint arXiv:2602.15397 (2026)
8. Gong, Z., Ding, P., Lyu, S., Huang, S., Sun, M., Zhao, W., Fan, Z., Wang, D.: Carp: Visuomotor policy learning via coarse-to-fine autoregressive prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13460–13470 (2025)
9. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: *pi*_{0.5}: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
10. Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., Sadigh, D.: Prismatic vlms: Investigating the design space of visually-conditioned language models. In: Forty-first International Conference on Machine Learning (2024)
11. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
12. Lee, S., Wang, Y., Etukuru, H., Kim, H.J., Shafiullah, N.M.M., Pinto, L.: Behavior generation with latent actions. arXiv preprint arXiv:2403.03181 (2024)

13. Li, Q., Liang, Y., Wang, Z., Luo, L., Chen, X., Liao, M., Wei, F., Deng, Y., Xu, S., Zhang, Y., et al.: Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. arXiv preprint arXiv:2411.19650 (2024)
14. Liang, Z., Mu, Y., Ma, H., Tomizuka, M., Ding, M., Luo, P.: Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16467–16476 (2024)
15. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
16. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
17. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
18. Liu, Y., Zhang, S., Dong, Z., Ye, B., Yuan, T., Yu, X., Yin, L., Lu, C., Shi, J., Yu, L.J.T., et al.: Faster: Toward efficient autoregressive vision language action modeling via neural action tokenization. arXiv preprint arXiv:2512.04952 (2025)
19. Lyu, H., Chen, C., Xie, S., Wang, P., Chen, X., Zhang, S., Xu, C.: Omnisat: Compact action token, faster auto regression. arXiv preprint arXiv:2510.09667 (2025)
20. Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., Fox, D.: Mimicgen: A data generation system for scalable robot learning using human demonstrations. arXiv preprint arXiv:2310.17596 (2023)
21. Marafioti, A., Zohar, O., Farré, M., Noyan, M., Bakouch, E., Cuenca, P., Zakka, C., Allal, L.B., Lozhkov, A., Tazi, N., et al.: Smolvlm: Redefining small and efficient multimodal models. arXiv preprint arXiv:2504.05299 (2025)
22. Mete, A., Xue, H., Wilcox, A., Chen, Y., Garg, A.: Quest: Self-supervised skill abstractions for learning continuous control. Advances in Neural Information Processing Systems **37**, 4062–4089 (2024)
23. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747 (2025)
24. Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., et al.: Smolvla: A vision-language-action model for affordable and efficient robotics. arXiv preprint arXiv:2506.01844 (2025)
25. Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., et al.: Octo: An open-source generalist robot policy. arXiv preprint arXiv:2405.12213 (2024)
26. Wang, Y., Zhu, H., Liu, M., Yang, J., Fang, H.S., He, T.: Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. arXiv preprint arXiv:2507.01016 (2025)
27. Wang, Y., Zhang, Y., Huo, M., Tian, R., Zhang, X., Xie, Y., Xu, C., Ji, P., Zhan, W., Ding, M., et al.: Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. arXiv preprint arXiv:2407.01531 (2024)
28. Zhang, J., Chen, Y., Xu, Y., Huang, Z., Zhou, Y., Yuan, Y.J., Cai, X., Huang, G., Quan, X., Xu, H., et al.: 4d-vla: Spatiotemporal vision-language-action pretraining with cross-scene calibration. Advances in Neural Information Processing Systems **38**, 33914–33937 (2026)

29. Zhong, Y., Liu, Y., Xiao, C., Yang, Z., Wang, Y., Zhu, Y., Shi, Y., Sun, Y., Zhu, X., Ma, Y.: Freqpolicy: Frequency autoregressive visuomotor policy with continuous tokens. Advances in Neural Information Processing Systems **38**, 56493–56526 (2026)