LEARNING TEMPORALLY CONTEXTUALIZED LATENT ACTION SPACE FOR ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot policies are commonly trained to predict low-level action chunks from observations. However, raw action spaces are weakly aligned with task progress and lack inherent temporal semantics. In this work, we propose an action representation learning framework that learns a Temporally Contextualized Latent Action (TCLA) space. Beyond action reconstruction, we introduce two pretraining objectives: a temporal forward prediction objective in latent space and a task-progress estimation objective using episode-level spectral action context. Through experiments, we show that our method consistently improves training efficiency, success rates, and motion smoothness across pretrained Vision-Language-Action (VLA) models and World Action Models (WAMs), as well as policies trained from scratch. For example, on a RoboTwin task set, TCLA improves X-VLA from 55.6% to 89.0% in just 10k steps, and improves Fast-WAM from 20.6% to 40.8% at epoch 5. It also provides a new perspective on extending single-task policies to multi-task settings, highlighting the broader potential of temporal action pretraining.

1 INTRODUCTION

Learning robot manipulation policies from demonstrations has made rapid progress with the development of visuomotor policies(Zhao et al., 2023; Chi et al., 2025), Vision-Language-Action (VLA) models(Kim et al., 2024; Black et al., 2025a; Community, 2026) and World Action Models (WAM)(Cen et al., 2025; Yuan et al., 2026; Liu et al., 2026c). Most existing methods directly predict low-level robot actions, such as joint positions or end-effector poses. This end-to-end framework is simple and effective, but requires policies to operate directly in low-level action spaces that may be poorly aligned with task-relevant behavioral structure.

Prior studies have shown that the choice of action representation can substantially affect policy learning(Allshire et al., 2021; Feng et al., 2026; Lee et al., 2024). However, existing action representations mainly focus on restructuring actions (Mete et al., 2024; Zhu et al., 2025; Liu et al., 2026a), with temporal context and task intention less explicitly modeled. It remains an open question how to model temporal and intention-aware structure in action representations and whether it benefits policy learning.

With practice, human motor execution gradually becomes less dependent on explicit, attention-demanding control and increasingly relies on automatized sensorimotor processes (Diedrichsen & Kornysheva, 2015; Krakauer et al., 2019). This phenomenon, often associated with muscle memory, suggests that practice induces a task-specific structured prior over the raw action space, organizing temporally related actions into compact and coherent representations. Such a prior can further mitigate the entanglement between semantic reasoning and low-level motion control during policy learning. Motivated by these insights, we argue that an intention-aware and temporally contextualized action space can benefit robot policy learning.

To this end, we introduce TCLA, a simple and lightweight framework for learning an intention-aware and temporally contextualized continuous latent action space. Beyond action reconstruction, TCLA introduces two complementary pretraining objectives. First, we formulate temporal forward prediction as flow matching between current and subsequent action latents. This objective structures the latent space by temporal continuity, organizing related actions into coherent representations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

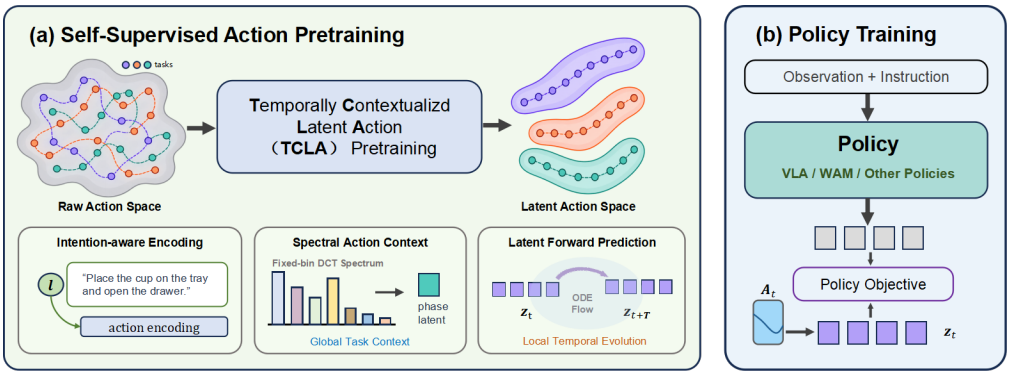


Figure 1: **Overview of TCLA.** (a) TCLA pretrains a structured latent action space by jointly incorporating task intention, global task context, and local temporal evolution, transforming raw action representations into an intention-aware and temporally contextualized space. (b) The pretrained action encoder provides latent action targets for downstream policy learning.

Second, we introduce a task-progress estimation objective that leverages episode-level spectral action context to capture the global temporal structure of task execution. The phase representation conditions both latent forward prediction and action decoding. By jointly modeling local temporal evolution and global task context, these objectives effectively encode multimodal action distributions and shape the latent action space into a more compact and organized structure.

We evaluate the proposed framework across pretrained Vision-Language-Action models, World Action Models, and policies trained from scratch on simulated and real-world manipulation tasks. TCLA accelerates convergence while improving task success. On the RoboTwin(Chen et al., 2025a) benchmark, X-VLA with TCLA reaches 89.0% at 10k steps, compared with 55.6% for raw-action prediction, and its best score increases from 82.8% to 91.8%. On the RoboTwin Hard set, the best X-VLA score increases from 59.0% to 71.8%, demonstrating improved robustness of the learned latent action representation. TCLA extends a single-task ACT policy to multi-task learning: a single ACT+TCLA policy trained for 50k steps reaches a ten-task average of 75.6%, compared with 61.0% from ten separate 50k single-task ACT trainings. In real-world experiments, TCLA raises the mean success rate from 43% to 78% and reduces overall RMS jerk from 0.012 to 0.007, producing smoother trajectories.

Our contributions are summarized as follows:

- We propose an action representation pretraining, combining global phase modeling with spectral action context and latent temporal forward prediction to learn an intention-aware and temporally contextualized latent action space.
- We systematically investigate how temporally structured action representations affect policy learning, showing consistent benefits in training efficiency, task success and motion smoothness.
- We conduct extensive experiments across simulated and real-world manipulation tasks, covering pretrained VLA models, WAMs, and policies trained from scratch, demonstrating the broad applicability of TCLA framework.

2 METHOD

We present an action representation learning framework that learns a Temporally Contextualized Latent Action (TCLA) space. Action representation pretraining is decoupled from downstream policy learning, allowing the same pretrained latent action space to be reused across different robot policies.

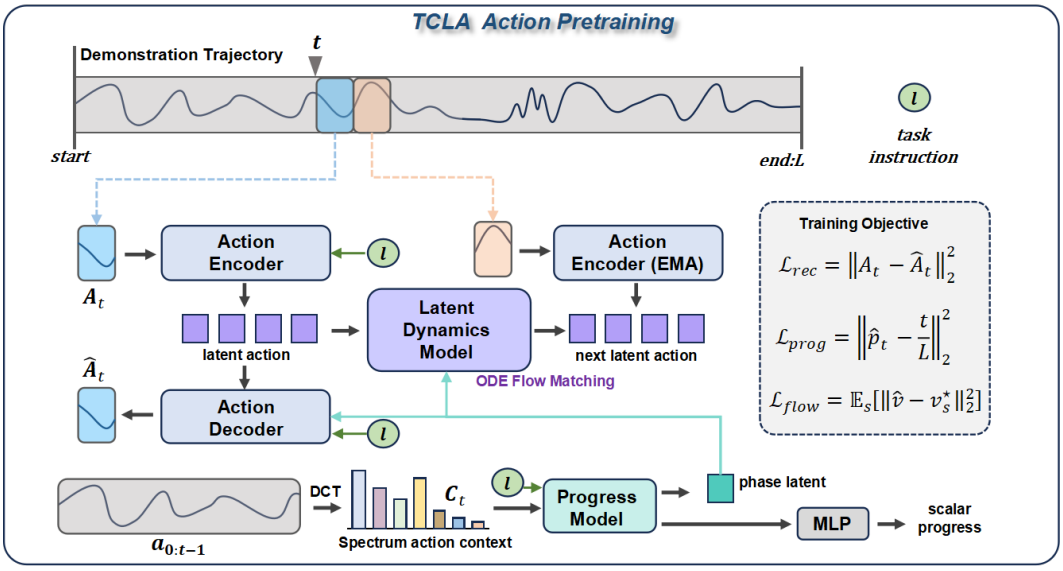


Figure 2: Action pretraining pipeline of TCLA. TCLA jointly learns latent actions through action reconstruction, latent temporal forward prediction with an EMA target encoder, and task-progress modeling from spectral action context.

2.1 Problem Formulation and Overview

We consider demonstration trajectories paired with task instructions. ℓ denotes the corresponding task instruction encoded by MiniLM(Wang et al., 2020). At time t, an action chunk of length T is denoted as

$$A_t = [a_t, \ldots, a_{t+T-1}] \in \mathbb{R}^{T \times D_a}. \tag{1}$$

Instead of directly learning policies over A_t, TCLA introduces a continuous multi-token latent action

$$z_t = E_\theta(A_t, \ell) \in \mathbb{R}^{Z \times D_z}, \tag{2}$$

where E_θ is the action encoder. Z denotes the latent action length, typically obtained with $4\times$ temporal downsampling ($Z = T/4$).

We detail the action pretraining framework in Sec. 2.2 and describe the policy training with TCLA in Sec. 2.3.

2.2 Temporally Contextualized Action Representation Learning

We first learn a continuous action representation through autoencoding. Given z_t, task instruction ℓ, and phase latent representation φ_t, the action decoder reconstructs the original action chunk:

$$\hat{A}_t = D_\theta(z_t, \ell, \varphi_t). \tag{3}$$

We optimize the reconstruction objective to preserve fine-grained action details:

$$\mathcal{L}_{\text{rec}} = \left\| A_t - \hat{A}_t \right\|_2^2 \tag{4}$$

We next describe how the phase latent representationφ_t is learned from historical action context.

Phase Modeling with Spectral Action Context Many works have explored task-progress encoding(Yan et al., 2026; Tan et al., 2025) and causally structured action decoding(Black et al., 2025b; Ma et al., 2025). Within the TCLA framework, we propose a new approach that maintains these properties within the action space.

The length of the executed actions grows continuously, making it difficult to obtain a fixed-shape representation. We transform the historical action sequence $\mathcal{H}_t = [a_0, \ldots, a_{t-1}]$ into the frequency domain using the Discrete Cosine Transform (DCT)(Khayam, 2003), and retain the first N frequency coefficients, compressing the variable-length history into a fixed number of frequency cosine coefficients ordered by DCT order.

$$C_t = \mathrm{Trunc}_N\left(\mathrm{DCT}(\mathcal{H}_t)\right) \in \mathbb{R}^{N \times D_a}, \tag{5}$$

We then encode the spectral action context together with the task instruction into a phase latent φ_t:

$$\varphi_t, \hat{p}_t = G_\theta(C_t, \ell), \tag{6}$$

where G_θ denotes a Perceiver-style(Jaegle et al., 2021) progress model. φ_t conditions both action reconstruction and latent temporal prediction. $\hat{p}_t$ is the predicted normalized episode progress. The corresponding progress objective is

$$\mathcal{L}_{\mathrm{prog}} = \left\| \hat{p}_t - \frac{t}{L} \right\|_2^2. \tag{7}$$

We use a phase latent representation rather than scalar episode progress. A scalar progress value captures only coarse temporal location. In contrast, the phase latent can retain richer information about temporal consistency and phase-dependent action multimodality, both of which are important for action generation (Liu et al., 2025; Lee et al., 2024).

Latent Temporal Forward Prediction Given the current action chunk A_t and a subsequent action chunk A_{t+T}, we encode them with the online encoder and an EMA target encoder (Grill et al., 2020), respectively, using the latter to provide slowly evolving targets for stable temporal representation learning:

$$z_t = E_\theta(A_t, \ell), \qquad \bar{z}_{t+T} = \bar{E}_{\bar{\theta}}(A_{t+T}, \ell). \tag{8}$$

We introduce a latent prediction model F_θ to model the direct transition from the current latent action z_t to the subsequent $\bar{z}_{t+T}$, conditioned on the phase representation φ_t. Similar to VITA (Gao et al., 2025) and A2A (Jia et al., 2026), we adopt an ODE-based flow-matching formulation. The flow directly transports between the two structured action latents rather than starting from random noise. For $s \sim \mathcal{U}(0, 1)$, we define

$$\tilde{z}_s = (1 - s)z_t + s\bar{z}_{t+T}, \qquad v_s^\star = \bar{z}_{t+T} - z_t. \tag{9}$$

The phase-conditioned dynamics model predicts the corresponding velocity field with the objective

$$\mathcal{L}_{\mathrm{flow}} = \mathbb{E}_s \left[\| F_\theta(\tilde{z}_s, s, \varphi_t) - v_s^\star \|_2^2 \right]. \tag{10}$$

This objective explicitly introduces local temporal structure into the action representation space. Rather than treating individual action chunks independently, it organizes temporally related actions into coherent latent transitions. Conditioning on φ_t further makes these transitions aware of their global task context. In general, this process temporally mirrors the model inference process, thereby providing a more suitable action representation for subsequent policy learning.

Joint Pretraining Objective The complete TCLA objective is

$$\mathcal{L}_{\mathrm{TCLA}} = \lambda_r \mathcal{L}_{\mathrm{rec}} + \lambda_f \mathcal{L}_{\mathrm{flow}} + \lambda_p \mathcal{L}_{\mathrm{prog}}. \tag{11}$$

The three objectives respectively preserve fine-grained action fidelity, model local temporal evolution, and capture global task context. Together, they shape the latent action space into a compact and organized representation. In experiments in this work, we set the three loss weights to 1.

2.3 Policy Learning and Inference with TCLA

Policy Learning. After pretraining, TCLA serves as the action prediction space for downstream policies. Given observation history O_H and task instruction ℓ, the policy predicts

$$\hat{z}_t = \pi_{\theta_\pi}(O_H, \ell), \qquad z_t = E_{\theta_E}(A_t, \ell). \tag{12}$$

The original policy objective can be retained by replacing raw action targets with latent actions. The target latents z_t can be precomputed and cached, introducing no additional encoder computation during policy training; they can also be computed online with the lightweight encoder. TCLA also allows the same pretrained latent action space to be reused across different policy architectures with minimal modification.

Inference. At execution time, the phase representation is inferred from action history and the predicted latent is decoded into executable actions:

$$\varphi_t = G_\theta^{\text{phase}}(C_t, \ell), \qquad \hat{A}_t = D_\theta(\hat{z}_t, \ell, \varphi_t). \tag{13}$$

During inference, this process unfolds recurrently, allowing the action decoder to maintain both local temporal structure and global task context, thereby promoting more consistent and smooth behavior.

3 Experiments

We design experiments to answer the following questions: **(Q1)** Does TCLA improve task success rates compared with raw-action prediction and task-agnostic action representations? **(Q2)** Does TCLA accelerate policy training convergence and improve the final optimization level? **(Q3)** Does TCLA improve motion smoothness and reduce unstable action changes? **(Q4)** In multi-task settings, can TCLA enable policies originally designed for single-task learning to effectively learn across multiple tasks?

3.1 Experimental Setup

Benchmarks and tasks. We evaluate TCLA on simulation benchmarks and real-world manipulation tasks. For RoboTwin, we select a subset of tasks that are suitable for action-space and action-representation comparison, following the spirit of recent action-space design studies(Feng et al., 2026). The selected tasks include *Adjust Bottle*, *Dump Bin Bigbin*, *Grab Roller*, *Lift Pot*, *Move Playingcard Away*, *Open Laptop*, *Place Burger Fries*, *Place Container Plate*, *Press Stapler*, and *Shake Bottle Horizontally*. For real-world evaluation, we test TCLA on five manipulation tasks: *Turn on Lamp*, *Press Buttons in Sequence*, *Stack bowls*, *Separate bowls* and *Insert Screw*.

Baselines. We evaluate TCLA with five baseline policies, X-VLA(Zheng et al., 2026), GO-1(Bu et al., 2025), Fast-WAM(Yuan et al., 2026), BAKU(Haldar et al., 2024), and ACT(Zhao et al., 2023), and compare those trained in the raw action space with policies trained with TCLA in the latent action space. BAKU is a transformer-based policy designed for multi-task manipulation, using a multimodal transformer encoder and a decoupled action head to generate action chunks. We instantiate its action head with a multilayer perceptron and train BAKU from scratch, which provides a from-scratch setting for evaluating TCLA.

X-VLA(Zheng et al., 2026) is a flow-matching-based vision-language-action policy that uses modality-specific encoders to align heterogeneous inputs before processing them with transformer blocks.

GO-1(Bu et al., 2025) is a vision-language-latent-action framework that combines a pretrained vision-language model with a mixture-of-experts policy to connect high-level task reasoning with low-level action generation. For both X-VLA and GO-1, we initialize the policy from pretrained weights and fine-tune it to predict the TCLA latent-action targets.

Fast-WAM(Yuan et al., 2026) is a World Action Model that combines a pretrained video diffusion transformer with an action expert and uses video co-training to learn world representations while dropping explicit future imagination at inference time. We include Fast-WAM to examine whether TCLA can transfer to a WAM-style policy architecture.

ACT(Zhao et al., 2023) is a visuomotor policy with a transformer encoder-decoder trained as a conditional variational autoencoder. We use ACT as a representative single-task policy and further evaluate whether TCLA can adapt this policy to a shared multi-task setting.

Table 1: RoboTwin evaluation on the ten-task benchmark. We report the ten-task average success rate (SR, %). BAKU is trained from scratch, whereas X-VLA, GO-1, and Fast-WAM are fine-tuned from pretrained policies. The best score within each method and variant is shown in bold.

From-scratch training

Method	Variant	50k	100k	150k	200k	250k	300k	350k
BAKU	Baseline	29.4	43.2	43.6	46.2	45.0	46.2	**46.4**
	TCLA	68.6	65.4	66.4	68.0	**71.4**	69.0	69.8

VLA fine-tuning

Method	Variant	10k	20k	30k	40k	50k	60k	70k
X-VLA	Baseline	55.6	61.0	67.8	67.8	77.2	**82.8**	81.8
	TCLA	89.0	90.4	89.0	88.2	90.2	90.6	**91.8**
GO-1	Baseline	60.8	64.0	57.4	70.8	**75.8**	75.6	74.0
	TCLA	78.2	**81.4**	79.6	77.8	80.0	78.2	81.0

WAM fine-tuning

Method	Variant	Epoch 5	Epoch 10	Epoch 15	Epoch 20	Epoch 25
Fast-WAM	Baseline	20.6	35.4	37.2	**42.2**	35.8
	TCLA	40.8	51.8	52.6	**54.8**	51.4

Metrics. We report task success rate (SR) as the primary metric. For the simulation benchmark, each policy is evaluated with 50 trails per task at a given checkpoint. For the real-world experiment, 20 trials are performed for each task.

To evaluate motion smoothness, we also report jerk-based metrics. Following prior smoothness evaluations(Li et al., 2026; Valle et al., 2025; Liu et al., 2026b), jerk is computed as the third-order finite difference of the executed trajectory:

$$j_t = \frac{x_t - 3x_{t-1} + 3x_{t-2} - x_{t-3}}{\Delta t^3}, \tag{14}$$

where x_t denotes the end-effector state or action trajectory at time (t). For simulation, we report the mean jerk over time and evaluation episodes, whereas the real-world experiments use RMS jerk.

3.2 Simulation Results

We first compare task execution performance on the RoboTwin benchmark. The selected tasks cover precise positioning, contact-rich manipulation, object reorientation, and long-horizon execution.

Success Rates. Table 1 reports the ten-task average success rate at each recorded checkpoint. Because BAKU is trained from scratch whereas X-VLA, GO-1 and Fast-WAM are fine-tuned from pretrained policies, all comparisons are made within the same training paradigm. Within the from-scratch setting, TCLA improves BAKU's best success rate from 46.4% to 71.4%. For pretrained fine-tuning, TCLA improves the best success rate by 9.0 percentage points for X-VLA and by 5.6 percentage points for GO-1. The same trend extends to Fast-WAM, where TCLA improves the success rate from 42.2% to 54.8%. These consistent improvements demonstrate the effectiveness of TCLA in both VLA and WAM.

Motion Smoothness. We further evaluate motion smoothness using the mean jerk of the executed simulation trajectories, where lower values indicate smoother temporal changes in the predicted actions. We compute mean jerk (rad/command^3) in each policy's native trajectory representation, excluding gripper channels.

TCLA reduces the mean jerk for all three policies. For X-VLA, the jerk reduced from **0.0085** to **0.0037**. For GO-1, the jerk drops from **0.0304** to **0.0162**. The jerk of Fast-WAM drops from **0.0707** to **0.0450**. The reduction across all three policies indicates that TCLA yields temporally smoother

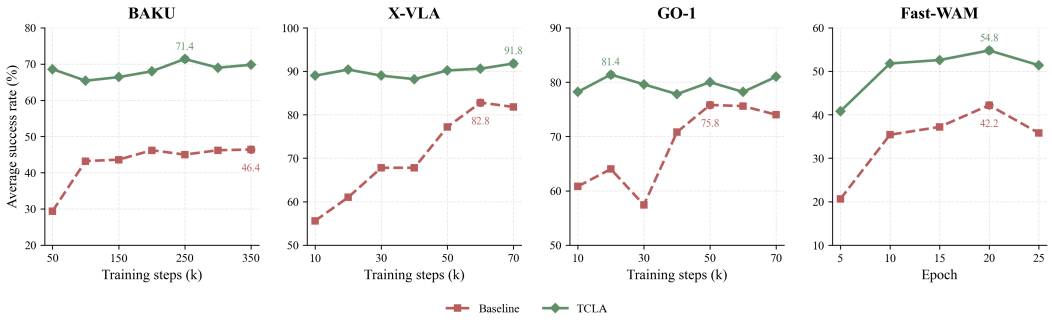


Figure 3: RoboTwin training convergence and sample efficiency. Each panel compares baseline and TCLA for one method.

action trajectories, consistent with the learned latent action space that captures local motion evolution and provides a more continuous prediction target than direct raw-action regression.

Convergence rate. We next examine how task performance evolves with the training step. Figure 3 shows the ten-task average success rate from RoboTwin for BAKU, X-VLA, GO-1 and Fast-WAM. The curves show an early advantage for TCLA across all methods. For X-VLA, TCLA reaches 89.0% at 10k steps, compared with 55.6% for baseline at the same checkpoint. TCLA reaches its best observed score of 91.8% at 70k steps, while baseline reaches 82.8% at 60k steps. The same trend appears for GO-1, where TCLA obtains 78.2% at 10k steps and its best score reaches 81.4%, compared with 60.8% and 75.8% for baseline. In the training from scratch setting, TCLA remains above baseline at all recorded steps and increases the best score from 46.4% to 71.4%.

These results indicate that TCLA provides a more effective training target in latent action space for both fine-tuning and from-scratch optimization. The improvement is most visible at early training stages, where TCLA reaches a higher success rate with fewer steps.

3.2.1 Generalization Ability

We evaluate generalization on the RoboTwin Hard test set, which introduces visual and geometric variations relative to the clean training environment, including scene clutter, randomized lighting and textures, and varying tabletop heights.

Table 2 reports the ten-task average success rate of X-VLA, GO-1 and Fast-WAM with and without TCLA at each recorded training checkpoint. For X-VLA, TCLA outperforms raw-action prediction throughout training, reaching the highest score of 71.8% at 40k steps, whereas the baseline reaches 59.0% only at 90k steps. GO-1 and Fast-WAM with TCLA outperform the baselines on the hard split as well. For Fast-WAM, both settings show lower success rates after Epoch 5. This consistent degradation may result from overfitting to the clean training environment, which can reduce performance under unseen visual conditions and after excessive optimization (Dai et al., 2022; Li et al., 2023). These results are consistent with the design choices of TCLA. Action reconstruction preserves fine-grained motion information, while latent temporal prediction and task-progress learning encourage the latent space to encode local trajectory evolution and task-phase structure. Consequently, the policy can robustly rely on task-relevant motion and progress information when the visual appearance and geometry of the scene change, which helps maintain successful execution under the tested variations.

3.2.2 Multi-task Learning

We examine whether the task-aware and temporally contextualized action representation introduced in Section 2.2 supports learning multiple manipulation tasks within a single policy. Using ACT as the policy backbone, we compare two settings: ACT trained separately for each task with raw-action targets, and ACT trained across all tasks with TCLA. Table 3 reports the success rate for each RoboTwin task and the corresponding ten-task average. Although ACT achieves strong performance when trained separately for each task, this setting requires a distinct policy for every task. In contrast,

Table 2: RoboTwin-Hard results. We report the mean success rate (%) across ten tasks. X-VLA and GO-1 use checkpoints at 10k-step intervals, while Fast-WAM uses checkpoints at 5-epoch intervals.

VLA fine-tuning

Method	Variant	10k	20k	30k	40k	50k	60k	70k	80k	90k	100k
X-VLA	Baseline	24.2	40.8	29.0	27.0	51.4	46.0	40.2	46.0	**59.0**	46.0
	TCLA	62.8	68.8	66.8	**71.8**	69.8	66.4	71.0	68.8	71.4	68.8
GO-1	Baseline	6.0	5.8	2.0	5.2	5.2	4.2	5.0	6.4	**6.4**	6.4
	TCLA	10.2	10.8	13.4	15.4	14.8	**17.4**	16.2	16.6	16.2	15.8

WAM fine-tuning

Method	Variant	Epoch 5	Epoch 10	Epoch 15	Epoch 20	Epoch 25
Fast-WAM	Baseline	**4.0**	2.0	2.6	2.6	2.8
	TCLA	**6.2**	5.0	5.4	3.4	5.4

Table 3: Multi-task learning results on ten RoboTwin tasks. ACT trains one single-task policy per task for 50k steps, whereas ACT+TCLA trains one multi-task policy for 50k steps in total.

Task	ACT (single-task)	ACT+TCLA (multi-task)
Adjust Bottle	92.0	**100.0**
Dump Bin Bigbin	60.0	**92.0**
Grab Roller	78.0	**90.0**
Lift Pot	58.0	**84.0**
Move Playingcard Away	**44.0**	40.0
Open Laptop	62.0	**70.0**
Place Burger Fries	38.0	**70.0**
Place Container Plate	30.0	**70.0**
Press Stapler	**58.0**	48.0
Shake Bottle Horizontally	90.0	**92.0**
Average	61.0	**75.6**

ACT+TCLA uses a single shared policy across all tasks and achieves a higher ten-task average success rate of 75.6% than the separately trained ACT policies of 61.0%.

This result suggests that TCLA provides a general interface for adapting a single-task policy to multi-task learning by replacing the original raw-action target with a task-conditioned latent-action target learned by TCLA, while keeping the policy architecture unchanged. The observation encoder, policy backbone, and action-prediction module can therefore be reused, providing a lightweight way to extend a single-task policy to multiple tasks without redesigning the policy network.

3.3 Real-World Results

We evaluate X-VLA with and without TCLA on five real-world manipulation tasks covering object pick-and-place, deformable object manipulation, sequential object pressing, and fine-grained insertion.

Table 4 reports success counts out of 20 trials and policy RMS jerk. TCLA improves success on all five tasks, increasing the aggregate success count from 43/100 to 78/100. Turn on Lamp requires pulling a switch cord over a narrow distance: a short pull fails to activate the lamp, whereas an excessive pull can topple it. The raw-action policy often fails to grasp the cord accurately or pull it by the appropriate distance, while TCLA more reliably localizes the cord and executes a suitably sized pull. Press Buttons requires sequential execution because similar visual observations occur at different stages while different actions are required. TCLA achieves 20/20 success, compared with 3/20 for the raw-action baseline, indicating that the temporal context and task-progress representation help disambiguate actions at different stages. Insert Screw requires fine spatial precision for

Table 4: Real-world results on five manipulation tasks. Task-level SR is reported as the number of successful trials out of 20.

Method	Variant	Turn on Lamp		Press Buttons		Stack bowls		Separate bowls		Insert Screw	
		SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓	SR ↑	Jerk ↓
X-VLA	Baseline	9/20	0.017	3/20	0.010	16/20	0.013	14/20	0.009	1/20	0.011
	TCLA	**18/20**	0.006	**20/20**	0.006	**18/20**	0.008	**18/20**	0.007	**4/20**	0.005

Table 5: Ablation results on the ten-task RoboTwin benchmark. The complete BAKU-TCLA model is included as the reference.

Variant	Best SR (%)
BAKU-TCLA	**71.4**
w/o EMA update	60.2
w/o Task progress	53.8
w/o Latent prediction	11.6
w/o Latent prediction and task progress	41.6

aligning and inserting the screw. The improvement from 1/20 to 4/20 suggests that TCLA provides a more precise action target for this fine-grained manipulation task. TCLA also reduces policy RMS jerk in both the interior of action chunks and at chunk transitions. This smoother behavior is consistent with the latent temporal prediction objectives, which encourage the latent action space to form smooth and temporally continuous trajectories for local action evolution. These results show that the benefits of the latent action representation extend to physical tasks, with improved task completion rate and smoother policy outputs.

3.4 Ablation Studies

Table 5 reports the ablations of TCLA based on BAKU baseline. The tasks are consistent with those in the RoboTwin simulation experiments. The variants remove the EMA update, task-progress objective, latent prediction objective, or both objectives. Across the reported results, the complete BAKU-TCLA model achieves the highest success rate, confirming the effectiveness of combining reconstruction with the two auxiliary objectives and EMA target updates. Removing the EMA update and task-progress prediction reduces the best success rate to 60.2% and 53.8% respectively. The variant without latent prediction performs substantially worse, reaching only 11.6%, suggesting that the temporal forward-prediction objective is important for learning the latent action representation. The result for the variant that removes both latent prediction and task-progress prediction is 41.6%. This indicates an interaction between the objectives rather than independent additive effects. When both auxiliary objectives are removed, the model is reduced to a simpler reconstruction-based autoencoder. This simpler optimization problem achieves a higher score in this ablation, although it remains substantially below the complete BAKU-TCLA model.

4 Conclusion

We presented TCLA, a novel framework for learning an intention-aware and temporally contextualized latent action space. Our results show that temporal pretraining of action representations, a relatively underexplored dimension in robot policy learning, can substantially improve downstream policy training. Across VLAs, WAMs, and policies trained from scratch, TCLA consistently improves training efficiency, task success, and motion smoothness, highlighting the importance of structuring the action space itself for robot learning.

AI Use Statement

In this work, we used generative AI tools to assist with language editing, including improving grammar, clarity, and presentation, as well as refining textual descriptions and figure layouts. We did not

use generative AI tools to generate experimental results, conduct data analysis, or make scientific claims. All AI-assisted content was carefully reviewed and verified by the authors. In particular, technical descriptions were checked against the actual methodology and implementation, and all reported results, analyses, and conclusions were independently validated by the authors. We take full responsibility for the final content of this work, including any text or artifacts produced with the assistance of generative AI.

Reproducibility Statement

We have made efforts to ensure the reproducibility of this work. The proposed TCLA framework, including its architecture and training objectives, is described in Section 2 and the Appendix B. Experimental settings, evaluation protocols, and implementation details are provided in Section 3 and the Appendix D. We also report the evaluation protocols used for the compared methods and apply consistent evaluation settings across baselines to facilitate fair and reproducible comparisons. The Appendix further includes model configurations, hyperparameters, and task details.

References

Arthur Allshire, Roberto Martín-Martín, Charles Lin, Shawn Manuel, Silvio Savarese, and Animesh Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6650–6656. IEEE, 2021.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*, 2025a.

Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint arXiv:2512.05964*, 2025b.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

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

Yixiang Chen, Peiyan Li, Yan Huang, Jiabing Yang, Kehan Chen, and Liang Wang. Ec-flow: Enabling versatile robotic manipulation from action-unlabeled videos via embodiment-centric flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11958–11968, 2025b.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Tianhong Dai, Kai Arulkumaran, Tamara Gerbert, Samyakh Tukra, Feryal Behbahani, and Anil Anthony Bharath. Analysing deep reinforcement learning agents trained with domain randomisation. *Neurocomputing*, 493:143–165, 2022.

Jörn Diedrichsen and Katja Kornysheva. Motor skill learning between selection and execution. *Trends in cognitive sciences*, 19(4):227–233, 2015.

Yuchun Feng, Jinliang Zheng, Zhihao Wang, Dongxiu Liu, Jianxiong Li, Jiangmiao Pang, Tai Wang, and Xianyuan Zhan. Demystifying action space design for robotic manipulation policies. *arXiv preprint arXiv:2602.23408*, 2026.

Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. Vita: Vision-to-action flow matching policy. *arXiv preprint arXiv:2507.13231*, 2025.

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

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning. *Advances in Neural Information Processing Systems*, 37:141208–141239, 2024.

Yuhan Hu, Hugues Thomas, Peide Huang, Mouli Sivapurapu, Benoit Landry, and Arto Kivila. Momo: Dial motion mode in robot manipulation with spatiotemporal action tokenization. *arXiv preprint arXiv:2607.26315*, 2026.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. *arXiv preprint arXiv:2602.07322*, 2026.

Miracle Kang, Lights Shi, Lucy Liang, Roy Gan, Dongxiu Liu, Pushi Zhang, Sylas Chen, Shawn Qin, Yinan Zheng, Jinliang Zheng, et al. X-tokenizer: A multimodal action tokenizer for vision-language-action pretraining. *arXiv preprint arXiv:2606.14752*, 2026.

Siddharth Karamcheti, Megha Srivastava, Percy Liang, and Dorsa Sadigh. Lila: Language-informed latent actions. In *Conference on robot learning*, pp. 1379–1390. PMLR, 2022.

Syed Ali Khayam. The discrete cosine transform (dct): theory and application. *Michigan State University*, 114(1):31, 2003.

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

John W Krakauer, Alkis M Hadjiosif, Jing Xu, Aaron L Wong, and Adrian M Haith. Motor learning. *Comprehensive physiology*, 9(2):613–663, 2019.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

Jiashun Li, Xiaoyu Shi, Hong Xie, Mingsheng Shang, and Yun Lu. Smoothvla: Aligning vision-language-action models with physical constraints via intrinsic smoothness optimization. *arXiv preprint arXiv:2603.13925*, 2026.

Qiyang Li, Aviral Kumar, Ilya Kostrikov, and Sergey Levine. Efficient deep reinforcement learning requires regulating overfitting. In *International Conference on Learning Representations*, 2023.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. Oat: Ordered action tokenization. *arXiv preprint arXiv:2602.04215*, 2026a.

Yuejiang Liu, Jubayer Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, volume 2025, pp. 4594–4627, 2025.

Yufeng Liu, Hang Yu, Juntu Zhao, Bocheng Li, Di Zhang, Mingzhu Li, Wenxuan Wu, Yingdong Hu, Junyuan Xie, Junliang Guo, et al. Learning native continuation for action chunking flow policies. *arXiv preprint arXiv:2602.12978*, 2026b.

Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. Oa-wam: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026c.

Jiahua Ma, Yiran Qin, Yixiong Li, Xuanqi Liao, Yulan Guo, and Ruimao Zhang. Cdp: Towards robust autoregressive visuomotor policy learning via causal diffusion. *arXiv preprint arXiv:2506.14769*, 2025.

Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control. *Advances in Neural Information Processing Systems*, 37:4062–4089, 2024.

Sangjun Noh, Dongwoo Nam, Kangmin Kim, Geonhyup Lee, Yeonguk Yu, Raeyoung Kang, and Kyoobin Lee. 3d flow diffusion policy: Visuomotor policy learning via generating flow in 3d space. *arXiv preprint arXiv:2509.18676*, 2025.

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

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Yuheng Ji, Cheng Chi, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, et al. Robo-dopamine: General process reward modeling for high-precision robotic manipulation. *arXiv preprint arXiv:2512.23703*, 2025.

Pablo Valle, Chengjie Lu, Shaukat Ali, and Aitor Arrieta. Evaluating uncertainty and quality of visual language action-enabled robots. *arXiv preprint arXiv:2507.17049*, 2025.

Shengjie Wang, Jiacheng You, Yihang Hu, Jiongye Li, and Yang Gao. Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation. *arXiv preprint arXiv:2501.14400*, 2025a.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11089–11099, 2025b.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim GJ Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

Li Wenjie, Yash Jangir, Ignacy Stepka, Yash Agarwal, Marion Kipsang, and Yonatan Bisk. Lost in reconstruction: Aligning action representations with language in vision-language-action models. *arXiv preprint arXiv:2608.10484*, 2026.

Rongtao Xu, Jian Zhang, Minghao Guo, Youpeng Wen, Haoting Yang, Min Lin, Jianzheng Huang, Zhe Li, Kaidong Zhang, Liqiong Wang, et al. A0: An affordance-aware hierarchical model for general robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13491–13501, 2025.

Hongyu Yan, Qiwei Li, Jiaolong Yang, and Yadong Mu. Progressvla: Progress-guided diffusion policy for vision-language robotic manipulation. *arXiv preprint arXiv:2603.27670*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

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

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Yihang Zhu, Weiqing Wang, Shijie Wu, Ye Shi, and Jingya Wang. Learning semantic atomic skills for multi-task robotic manipulation. *arXiv preprint arXiv:2512.18368*, 2025.

A Related Work

A.1 Structured Prediction Spaces for Robot Manipulation

A growing body of work explores structured prediction spaces beyond raw robot actions. PerAct (Shridhar et al., 2023) represents manipulation in a language-conditioned 3D voxel action space, while RVT (Goyal et al., 2023) and RVT-2 (Goyal et al., 2024) use multi-view heatmaps for efficient spatial action prediction. Other approaches introduce interaction-centric representations, including keypoints and affordances (Yuan et al., 2024; Wang et al., 2025a; Xu et al., 2025), or represent actions through point trajectories and dense flow fields (Wen et al., 2023; Chen et al., 2025b; Noh et al., 2025). These methods mainly emphasize spatial or interaction structure. TCLA instead organizes continuous latent actions by local temporal evolution and global task context, yielding an intention-aware and temporally contextualized prediction space.

A.2 Action Representation Learning

Prior works on learned action representations mainly follow two directions. One line learns skill-level or latent action abstractions, ranging from language-conditioned latent control spaces (Karamcheti et al., 2022) to reusable skill representations (Mete et al., 2024; Zhu et al., 2025). Another line studies action tokenization, including learned action codebooks (Lee et al., 2024; Wang et al., 2025b), ordered action tokens (Liu et al., 2026a), and semantically aligned token representations (Kang et al., 2026; Wenjie et al., 2026). These methods improve abstraction, compactness, multimodality, or semantic organization, but task-specific temporal context is less explicitly modeled. Related works capture temporal structure from complementary perspectives. MoMo (Hu et al., 2026) disentangles task-dependent spatial structure from motion dynamics, while Temporal Straightening (Wang et al., 2026) regularizes latent temporal geometry for planning. TCLA instead directly organizes the latent action space according to local temporal evolution and global task context, making temporal contextualization an explicit objective of action representation learning.

B Architecture Details

TCLA consists of an online action encoder, an action decoder, a progress encoder, a latent dynamics model, and an EMA target encoder. Unless otherwise specified, we use an action chunk length of $T = 32$, latent dimension $D_z = 16$, temporal downsampling ratio $R = 4$, and $N = 24$ frequency bins. Each action chunk is therefore represented by $T/R = 8$ continuous latent tokens.

B.1 Action Encoder

The action encoder maps an action chunk $A_t \in \mathbb{R}^{32\times 8}$ and task instruction ℓ into eight 16-dimensional latent tokens. Actions and language are first projected to a hidden dimension of 256. The action sequence is temporally downsampled from 32 to 8 tokens by three causal convolution layers with kernel sizes $5, 3, 3$ and strides $2, 2, 1$, respectively.

The resulting action tokens are concatenated with one language token and processed by a two-layer Transformer encoder with four attention heads and an FFN dimension of 1024. The eight action tokens are finally projected from 256 to 16 dimensions to obtain $z_t \in \mathbb{R}^{8\times 16}$.

B.2 Action Decoder

The action decoder reconstructs the full action chunk from the latent action, task instruction, and phase representation. The 8 latent tokens, 1 language token, and 1 phase token are independently projected to a hidden dimension of 256 and form a 10-token decoder input.

The decoder uses 32 fixed sinusoidal action queries and two Transformer decoder layers with eight attention heads and an FFN dimension of 1024. No causal mask is applied between action queries, and the entire 32-step action chunk is decoded in parallel. During training, Gaussian noise with standard deviation 0.02 is added to the phase representation, together with phase dropout with probability 0.10.

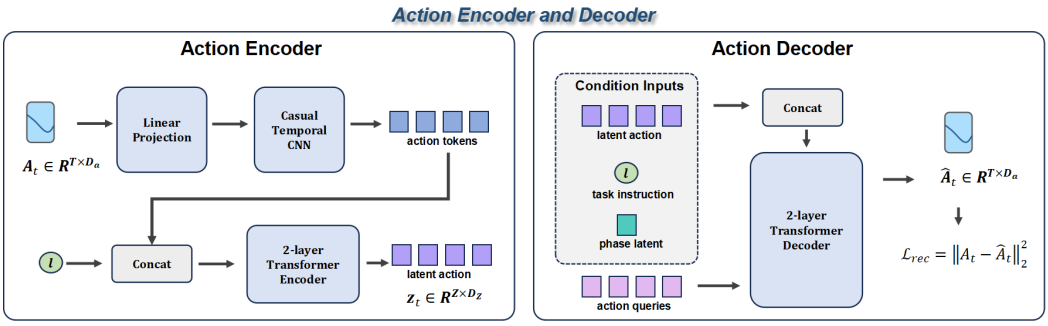


Figure 4: **Architecture of the TCLA action encoder and decoder.** The action encoder first projects and temporally compresses an action chunk A_t with a causal temporal CNN, and then combines the resulting action tokens with the task instruction ℓ through a Transformer encoder to produce the continuous latent action z_t. The action decoder conditions on z_t, the task instruction ℓ, and the phase latent φ_t, and uses a Transformer decoder with fixed action queries to reconstruct the action chunk $\hat{A}_t$. The reconstruction objective preserves fine-grained action information in the learned latent representation.

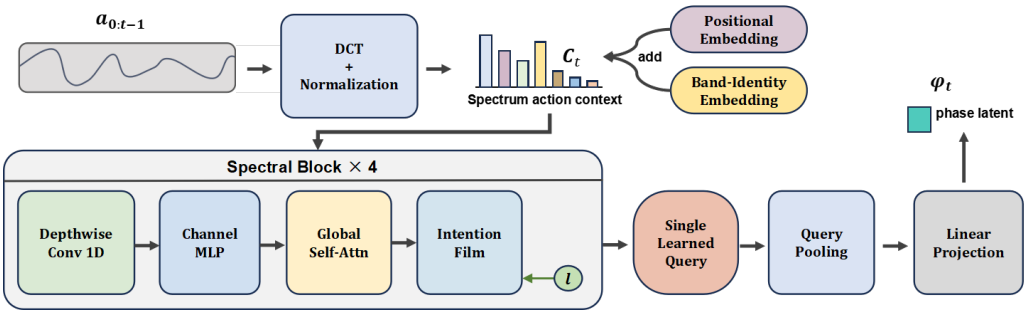


Figure 5: **Architecture of the TCLA progress encoder.** The executed action history is converted into a normalized spectral action context using DCT and augmented with positional and band-identity embeddings. Four language-conditioned spectral blocks model local and global frequency structure, after which learned-query pooling produces the phase latent φ_t representing the global task phase.

B.3 Progress Encoder

Given the executed action history, we first construct a fixed-length spectral context $C_t \in \mathbb{R}^{N \times D_a}$ using an orthogonally normalized DCT. Each frequency token is independently normalized using training-set statistics and projected into a hidden space:

$$h_k = W_v C_{t,k} + E_{\text{freq}}(k) + E_{\text{band}}(k), \tag{15}$$

where E_{freq} encodes the frequency index and E_{band} provides a coarse frequency-band identity.

The resulting spectral tokens are processed by four spectral blocks. Each block combines depthwise convolution along the frequency axis, a channel MLP, global multi-head self-attention, and language-conditioned feature modulation. The local convolution captures interactions between neighboring frequencies, while self-attention models long-range dependencies across the spectrum.

A single learned query then aggregates all spectral tokens through cross-attention, producing a compact history representation. Two prediction heads are applied to the pooled feature: a vector-valued phase representation φ_t and a scalar progress estimate $\hat{p}_t$. The former conditions both action decoding and latent dynamics, while the latter provides explicit supervision toward normalized episode progress.

B.4 Latent Dynamics Model

The latent dynamics model predicts the velocity field between the current action latent and the subsequent latent produced by the EMA target encoder. Each 16-dimensional latent token is projected to a hidden dimension of 256.

Flow time is represented using a sinusoidal embedding followed by an MLP, while the 16-dimensional phase representation is independently projected to the same hidden dimension. Their embeddings are combined to condition four MLP-based FlowNet layers through feature-wise modulation.

The dynamics model outputs

$$v_\theta \in \mathbb{R}^{8\times 16}.$$

The same conditional transformation is applied independently to each latent token. Cross-token temporal structure is therefore primarily encoded by the action encoder and jointly interpreted by the decoder.

B.5 EMA Target Encoder

The target encoder has the same architecture as the online action encoder and is initialized as its copy. Its parameters are not updated by gradient descent, but instead follow the online encoder through exponential moving average:

$$\bar{\theta}_E \leftarrow m\bar{\theta}_E + (1-m)\theta_E.$$

The target branch is evaluated without gradients and provides slowly evolving latent targets for temporal representation learning.

B.6 Model Size

Table 6 summarizes the parameter count of each component.

Table 6: Parameter counts of the TCLA action representation model.

Module	Total Params.	Trainable Params.
Action Encoder	2.484M	2.484M
Action Decoder	2.220M	2.220M
Latent Dynamics	3.497M	3.497M
Progress Encoder	1.102M	1.102M
EMA Target Encoder	2.484M	0
Total	11.787M	9.303M

C Task Details

Figure 6 visualizes the ten RoboTwin tasks used in the simulation experiments. The task set spans object placement, grasping, pouring, lifting, reorientation, pressing, and long-horizon object manipulation.

Simulation tasks. The details of 10 tasks selected from RoboTwin benchmark are described below.

Adjust Bottle requires the robot to grasp and reposition a bottle with controlled translation and orientation. The task tests precise object handling under a constrained tabletop workspace.

Dump Bin Bigbin requires grasping a container and pouring its contents into a target bin. Successful execution depends on coordinated transport and controlled object tilt.

Grab Roller requires grasping a cylindrical roller and moving it from its initial location. The task evaluates contact-aware grasping and motion around a compact object.

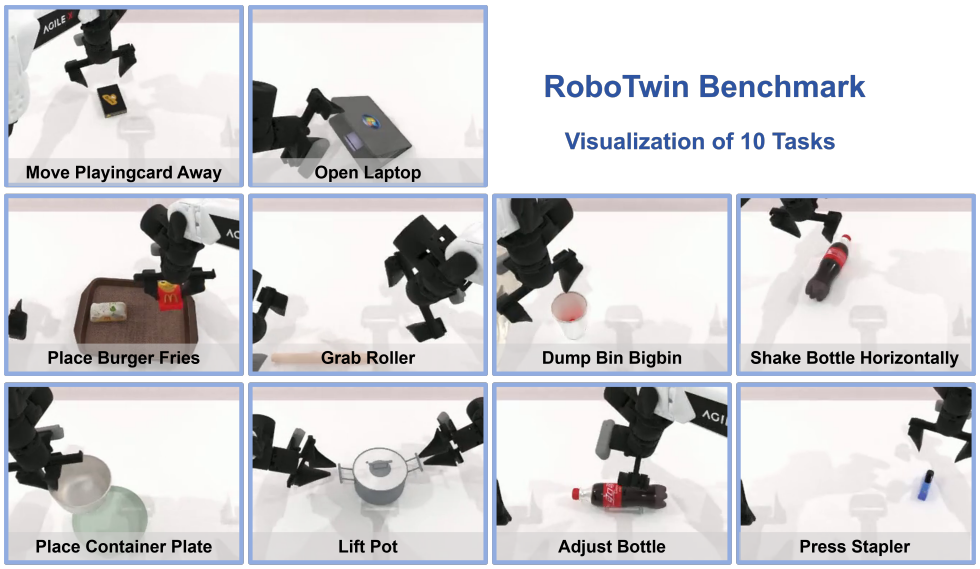


Figure 6: The ten RoboTwin tasks used in the simulation experiments.

Lift Pot requires grasping a pot and lifting it from the tabletop. The task emphasizes accurate approach, stable grasp formation, and vertical motion.

Move Playingcard Away requires moving a thin playing card away from its initial position. The small object and limited contact surface make precise end-effector positioning important.

Open Laptop requires manipulating a laptop lid from the closed state to an open state. The task involves locating a hinged object and applying a coordinated motion over a longer interaction.

Place Burger Fries requires placing food items into a tray or target region. The task combines object transport with accurate placement relative to the receptacle.

Place Container Plate requires moving a container and placing it onto a plate. The task tests spatial alignment and controlled release at the target location.

Press Stapler requires pressing a stapler at a specified location. The task involves contact-rich interaction and sufficient downward motion to trigger the mechanism.

Shake Bottle Horizontally requires grasping a bottle and executing a horizontal shaking motion. The task evaluates temporally extended motion generation with repeated coordinated movements.

Figure 7 shows the five real-world Franka tasks used for physical evaluation. These tasks cover precise cord manipulation, sequential button pressing, bowl stacking and separation, and fine-grained screw insertion.

Real-world tasks. We design five distinct table-top real-world manipulation tasks.

Turn on Lamp requires grasping a narrow switch cord and pulling it over a short, precise distance. The pull must be long enough to activate the lamp but controlled enough to avoid destabilizing the lamp.

Press Buttons in Sequence requires pressing a prescribed sequence of buttons whose visual appearance can be similar across stages. The task therefore tests whether the policy uses temporal context to select the stage-appropriate action.

Stack bowls requires grasping and placing one bottle on top of another. The task evaluates three-dimensional alignment, stable placement, and controlled release.

Separate bowls requires separating two initially adjacent bowls and moving them apart. Successful execution depends on precise grasping and collision-aware manipulation.

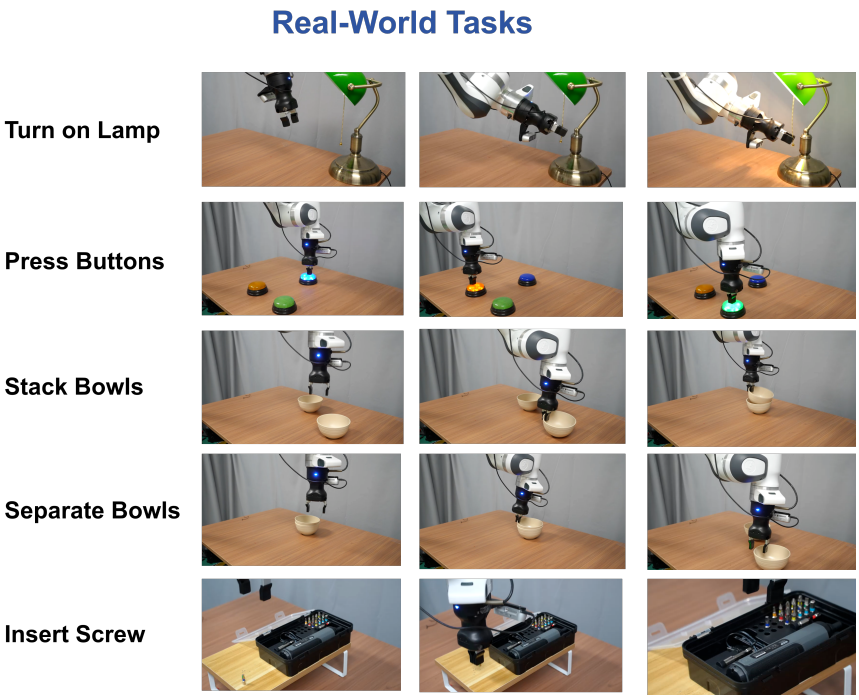


Figure 7: The five real-world Franka manipulation tasks used in the physical experiments.

Insert Screw requires aligning a screwdriver with a screw and inserting it into the target assembly. The task places a high demand on spatial precision and fine-grained end-effector control.

D TCLA Pretraining Details

TCLA is jointly optimized using action reconstruction, latent flow matching, and task-progress prediction:

$$\mathcal{L}_{\text{TCLA}} = \lambda_r \mathcal{L}_{\text{rec}} + \lambda_f \mathcal{L}_{\text{flow}} + \lambda_p \mathcal{L}_{\text{progress}}.$$

We optimize TCLA using AdamW with a learning rate of 10^{-4}, $\beta = (0.9, 0.95)$, weight decay 10^{-4}, and a constant learning-rate schedule without warmup. We use a batch size of 256 per process and no gradient accumulation. Unless otherwise specified, training is performed in full precision without gradient clipping. The reconstruction, flow, and progress objectives are equally weighted. The EMA target encoder is updated after every optimization step, with its momentum linearly increasing from 0.996 to 1.0.

E More Results

Figure 8 shows the early training dynamics of X-VLA and GO-1 on the RoboTwin benchmark.

For X-VLA, TCLA starts at 7.2% at 1k steps compared with 6.8% for the raw-action baseline, then rises to 50.4% at 2k steps and 89.0% at 10k steps, while the baseline reaches 55.6% at 10k steps. For GO-1, TCLA similarly improves from 32.2% to 78.2% , whereas the baseline increases from 7.6% to 60.8%. TCLA remains above the baseline at every recorded checkpoint for both policies, indicating that the latent action target provides a more effective optimization signal from the beginning of policy training. The early advantage is accompanied by rapid improvement: TCLA gains 81.8 percentage points for X-VLA and 46.0 percentage points for GO-1 over the plotted range, compared with 48.8 and 53.2 percentage points for the respective baselines.

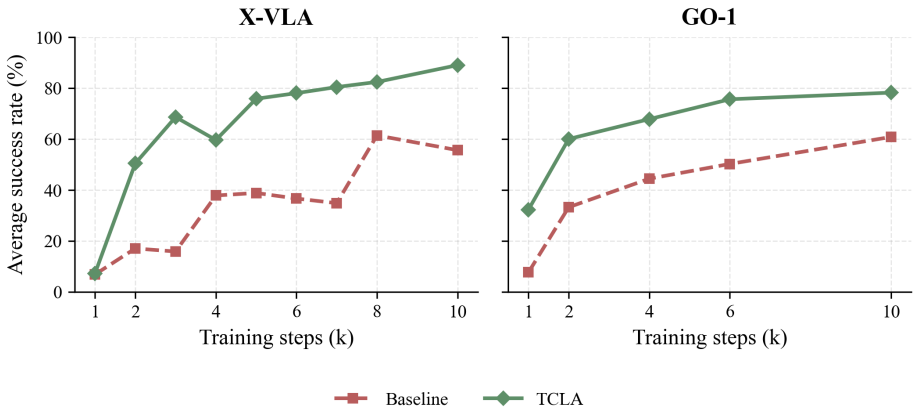


Figure 8: Early RoboTwin training convergence for X-VLA and GO-1.

E.1 Limitations and Future Work

Limitations. First, the scalability of a shared latent action space to substantially larger and more diverse task sets remains to be systematically studied. Second, the current formulation focuses on continuous latent actions and does not explore alternative discrete or hybrid representations. Third, the transferability of TCLA to unseen environments and robot embodiments has not yet been explored.

Future Work. Future work may explore discrete or VQ-based action representations, integrate TCLA with higher-level skill tokens or latent skills, and investigate transferable temporal motor priors learned from larger datasets that can be shared across related tasks and embodiments.