REACT: Responsive Execution-aware Action Coordination And Tuning For Moving Object Manipulation

1st Xiaofei Qin
School of Optical-Electronic and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, China
xiaofei.qin@usst.edu.cn

2nd Jianyu Zhang
School of Optical-Electronic and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, China
233370866@st.usst.edu.cn

3rd Anluo Yi
School of Optical-Electronic and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, China
232260517@st.usst.edu.cn

***Abstract*—Learning to reliably and stably manipulate moving objects with bimanual agents remains a challenging task. Many real-world manipulation tasks involve dynamic scenes, where objects of interest may be in motion. While existing visual manipulation methods primarily focus on manipulating static objects, leading to limited applicability when dealing with moving objects. These methods struggle to meet the strict demands on real-time responsiveness and robustness required in moving object manipulation. Furthermore, large-scale datasets for dynamic manipulation settings remain scarce. To alleviate these challenges, we introduce REACT (Responsive Execution-aware Action Coordination and Tuning for moving object manipulation), a novel sim-to-real method for manipulation of moving objects. Within REACT, we adopt a simulation-based dynamic scene dataset generation strategy for bimanual agents and employ a Slow–Fast policy to achieve closed-loop control and real-time refinement of action trajectories. This design enables zero-shot transfer from simulation to real-world dynamic settings, substantially improving stability and responsiveness in manipulation. Extensive experiments on real-world moving object manipulation tasks demonstrate that REACT outperforms existing visual imitation learning methods.**

***Index Terms*—Moving object manipulation, Reinforcement learning, Behavior cloning, Slow-Fast policy**

I. Introduction

Bimanual robots, as a key platform for embodied intelligence, are capable of performing numerous tasks that are infeasible for single-arm systems [1]. With the aid of large-scale demonstration data, visual imitation learning has achieved remarkable progress in static manipulation tasks for bimanual agents [2]. However, as illustrated in Fig. 1, most existing Bimanual methods [3], [4] remain dominated by open-loop or semi closed-loop paradigm. Open-loop methods execute actions without real-time perceptual feedback, while closed-loop methods continuously adjust actions based on incoming sensory information, offering stronger adaptability to environmental changes [5]. Yet, in real-world dynamic settings where targets move continuously and sensor delays are present, open-loop methods struggle to meet the need for real-time adjustment, severely limiting their practical applicability and reasoning performance.

Compounding this issue, the high complexity of controlling two arms and grippers makes the collection of human demonstrations for bimanual agents prohibitively costly, leading to a scarcity of high-quality dynamic interaction data [6]. This scarcity, in turn, makes it difficult to learn competitive policies for moving object manipulation solely from offline data or static simulations.

To alleviate these problems, we propose REACT ((Responsive Execution-aware Action Coordination and Tuning for moving object manipulation), a novel sim-to-real framework for closed-loop manipulation of moving objects. As shown in Fig. 2, the REACT framework operates by first automating the generation of large-scale dynamic scenes demonstration data in simulation [7] and then train a generative model, a conditional variational autoencoder (CVAE), on this dataset for behavior cloning [8]. Within this framework, we propose a closed-loop bimanual agents with a Slow-Fast Policy. The CVAE serves as the slow Policy, producing foundational action chunks for future motion sequences. A complementary Fast Policy then performs real-time, closed-loop refinement of these chunks. By leveraging semantic guidance from foundational action chunks and refining them in real-time with visual observations, our policy corrects for dynamic manipulation errors caused by perceptual latency in open-loop paradigms, thereby establishing a novel closed-loop paradigm. This paradigm enables successful zero-shot or near-zero-shot real-world deployment. Our contributions are summarized as follows:

This work was supported by the Shanghai Magnolia Talent Program Pujiang Project under [25PJD089].

- To support the manipulation of moving objects by bimanual agent, we adopt a large-scale simulation-based pipeline to generate dynamic scene demonstration data. By randomizing initial conditions, we produce a dataset with diverse moving target trajectories.
- We propose a sim-to-real framework including a Slow-Fast policy. The slow policy is responsible for global action sequencing and task planning, while the fast policy performs real-time adjustment and robust compensation at a fine time scale. Together, they form a fully closed-loop paradigm of bimanual agents.
- We demonstrate that REACT outperforms current visual imitation learning baselines in real-world benchmarks in terms of real-time capability and success rate. The results confirm that our approach enables zero-shot transfer to real-world dynamic scenes, opening a viable path toward the practical deployment of intelligent agents in dynamic scenes.

II. Related Works

A. Imitation Learning for Manipulation Policy

Dynamic scene tasks such as object manipulation and target following require policies that maintain stable perception and real-time trajectory updates. Imitation learning directly extracts operational patterns from expert demonstrations [11], avoiding complex model construction, and has thus become an effective approach for such dynamic scenes. Basic behavior cloning (BC) methods [12] treat policy learning as a supervised mapping, but are prone to state distribution shift in dynamic scene tasks. Improvements include incorporating historical information [13], using alternative architectures, modifying training objectives [14], and adding regularization [15]. To further capture temporal dependencies and manipulation intent, recent works employ hierarchical trajectory models [16] or latent action spaces [17] to model target motion and control sequences.

B. Reinforcement Learning with Simulation

Reinforcement learning (RL) enables the direct learning of adaptive and robust control policies through environment interaction and has demonstrated advantages in complex grasping and dynamic target tracking tasks. Leveraging high-fidelity simulators such as MuJoCo and Isaac Sim, researchers can train policies at scale in simulation and use the learned policies to generate diverse and scalable demonstration data, alleviating the high cost of real-world data collection. Works based on these simulation platforms typically employ offline RL, hybrid imitation + RL approaches, or examples generated from programmatic controllers [6], [10] to improve policy generalization. To reduce the sim-to-real (sim-to-real) gap, prior studies have explored domain randomization [18], invariant feature learning [19], and online adaptive fine-tuning, allowing simulated policies to maintain robust performance in real environments.

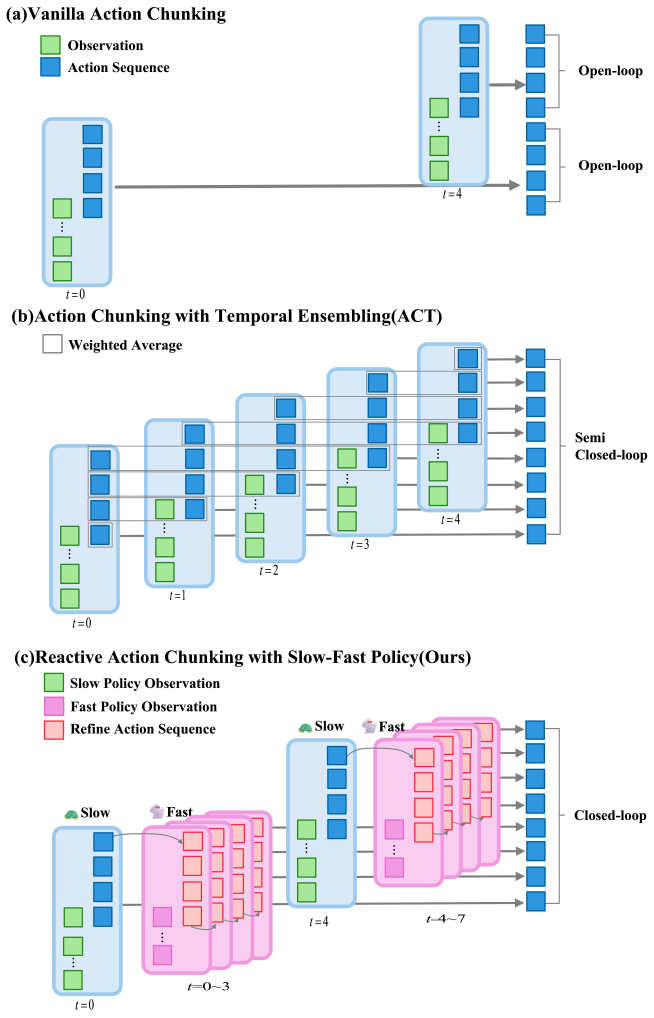

Fig. 1. **Comparison of three action chunking paradigms. (a)** Vanilla Action Chunking [9] adopts an open-loop paradigm, generating a 4-frame action chunk every 4 frames without intermediate feedback. **(b)** ACT [10] employs a semi-closed-loop paradigm, where action sequence are obtained by weighted averaging of low-frequency action chunks, yielding intermittent feedback. **(c)** Reactive Action Chunking with Slow-Fast Policy operates in a closed-loop paradigm, where a slow policy produces coarse action sequences and a fast policy refines them based on high-frequency observations.

C. Action Chunking

To mitigate error accumulation in long action sequences, action chunking methods are widely adopted. Diffusion Policy [20], ACT [10], and π_0 [21] employ action chunking to enhance temporal consistency and capture non-Markovian behaviors such as pauses or oscillatory motions present in demonstrations. However, these approaches often operate in an open-loop manner during execution, preventing policies from adjusting action sequences in response to environmental changes, thereby limiting reliability in dynamic scenes. Dynamic scene tasks require policies that can maintain stable control while adapting in real time. Recent methods introduce the CVAE into trajectory modeling [8], explicitly learning a structured latent space for future reference.

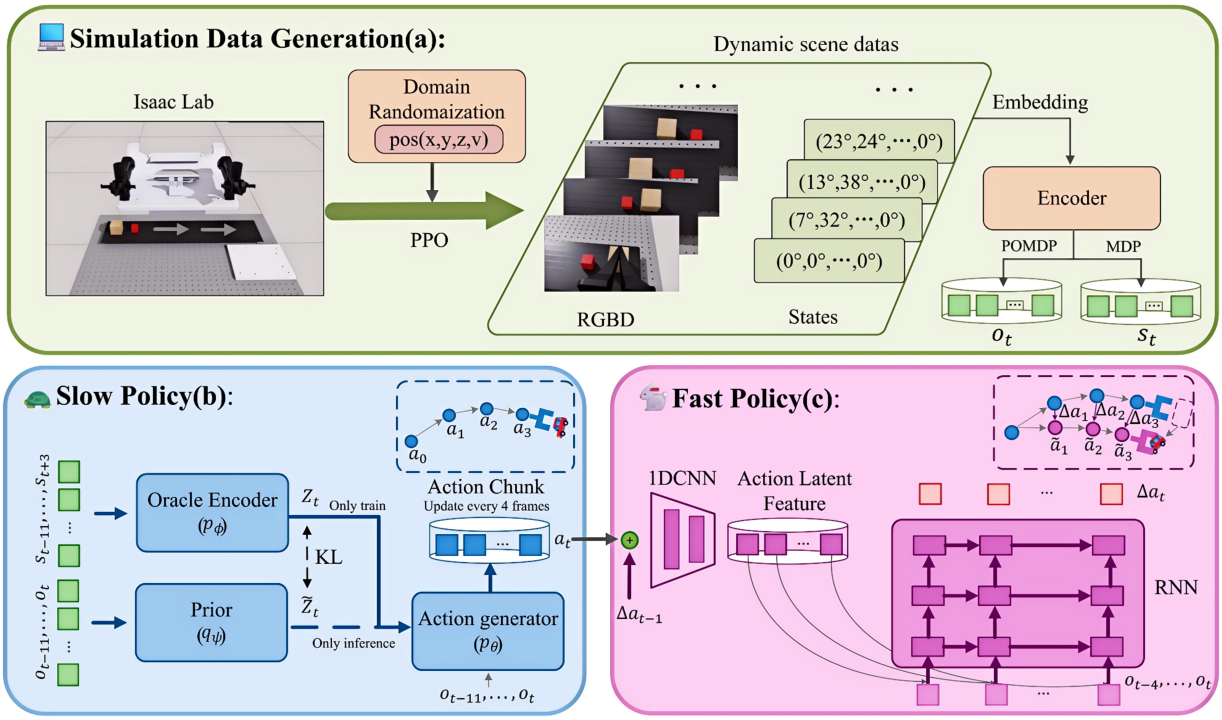

Fig. 2. **Overview of the proposed REACT.** (a) **Simulation and data generation.** We build parallel Isaac Lab environments with domain randomization (position and velocity variations) and collect dynamic RGB-D observations and proprioceptive states using PPO-driven rollouts. Encoders transform these signals into POMDP observations o_t and MDP states s_t. (b) **Slow Policy.** A variational inference module learns latent variables Z_t and $\widetilde{Z}_t$ from long-horizon observations and states through an oracle encoder p_ϕ and a prior network q_ψ, regularized by KL divergence. An action generator p_θ produces coarse action chunks updated every 4 frames. (c) **Fast Policy.** A 1D-CNN encoder and an RNN capture short-horizon visual dynamics to estimate a residual correction Δa_t, which is added to the slow policy output to obtain refined high-frequency actions $\tilde{a}_t$.

III. Method

A. Problem Formulation

Robotic manipulation is typically modeled as a Partially Observable Markov Decision Process (POMDP) [22], where the agent aims to learn a policy π mapping from observation-action histories to actions $a_t \sim \pi(a_t \mid \tau_t)$.

For fully observable Markov Decision Processes (MDPs), the agent has access to the complete state s_t, and the policy $\pi(a_t \mid s_t)$ is optimized to maximize the expected cumulative reward:

$$\arg\max_{\pi} \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\Big], \quad a_t \sim \pi(a_t \mid s_t) \tag{1}$$

where $r(s_t, a_t)$ is the immediate reward function and $\gamma \in (0, 1)$ is the discount factor.

In a POMDP, the agent cannot directly observe the full state s_t; instead, it receives partial information through an observation function$o_t = \mathcal{O}(s_t)$. The policy thus depends on the history of past observations and actions $\tau_t = (o_0, a_0, o_1, a_1, \ldots, a_{t-1}, o_t)$, so that the objective becomes:

$$\arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r(o_t, a_t)\Big], \quad a_t \sim \pi(a_t \mid \tau_t) \tag{2}$$

In practice, solving a POMDP requires maintaining distributions over all possible states, which is computationally expensive. Factors such as occlusion, sensor noise, and dynamic scene changes further amplify this complexity, making efficient decision-making under uncertainty a significant challenge for robotic manipulation.

B. Simulation Data Generation

Imitation learning from human demonstrations avoids complex system modeling and hyperparameter tuning [23], but is limited by the cost of collecting large-scale datasets. To mitigate this, we first train an expert policy as a data generator Fig.2 (a). In a physically consistent Isaac Lab simulator [24], we construct a simulation environment including the platform, objects, and an ALOHA bimanual robot equipped with three RGB cameras [10].

Leveraging the full observability of the simulation environment, the expert policy $\pi_\mu(a_t \mid s_t)$ can be formulated as a MDP, where the state s_t contains privileged information from the simulator. The policy is optimized using Proximal Policy Optimization (PPO) with the objective:

$$\arg\max_{\mu} \mathbb{E}_{\pi_\mu}\Big[\sum_{t} \gamma^t r_t\Big] \tag{3}$$

where γ is the discount factor. The reward r_t is decomposed into task, regularization, and penalty terms:

$$r_t = w_1(t) r_{\text{task}} + w_2(t) r_{\text{reg}} + w_3(t) r_{\text{penalty}} \tag{4}$$

with $w_1(t)$ kept constant, and $w_2(t), w_3(t)$ linearly increasing over time to balance early exploration and later stability [25], [26].

To improve data diversity and sim-to-real robustness, we introduce domain randomization:

$$\mathcal{D}_{\text{sim}} = \{\xi_{\text{asset}}, \xi_{\text{dyn}}\} \tag{5}$$

where ξ_{asset} perturbs object positions and sizes to prevent overfitting, and ξ_{dyn} introduces dynamic variations such as PD gains, sensor noise, and external forces [6].

Unlike conventional methods that process visual and action inputs separately, in this work we construct a joint observation by concatenating encoded visual observations and recent action sequences:

$$o_t = \text{concat}\Big(\text{Enc}_{\text{vis}}(\text{RGB}_t),\ \text{Enc}_{\text{act}}(a_{t-k:t})\Big) \tag{6}$$

where Enc_{vis} and Enc_{act} denote the visual and action encoders, respectively [8], [27]. The concatenated representation o_t forms a unified observation space, which serves as input to downstream policy learning.

C. Execution-aware Slow Policy

In a POMDP, robotic manipulation planning can be naturally formulated as a phased generation problem. In this paper, we model policy learning as a conditional generative problem with a prior distribution shown in Fig.2 (b), aiming to preserve the diversity and spatiotemporal consistency of expert policies under partial observability. Specifically, at each time step t, we assume the existence of a latent prior distribution $\tilde{z}_t$ that captures ambiguity in mapping observations to actions, as well as future trends. The conditional action distribution can then be expressed as:

$$\pi(a_t \mid o_t, h_t) = \int p_\theta(a_t \mid o_t, h_t, \tilde{z}_t)\, p_\phi(\tilde{z}_t \mid o_t, h_t)\, d\tilde{z}_t \tag{7}$$

where p_θ is an action generator and p_ϕ is a prior network that outputs $\tilde{z}_t$ based only on the observation o_t and history $h_t = \{o_0, \ldots, o_{t-1}\}$.

To enable effective learning of the prior distribution, we introduce a privileged encoder $q_\psi(z_t \mid s_t^{\text{oracle}})$ during training. This encoder leverages the privileged states $s^{\text{oracle}} = \{s_{t-11:t+3}^{\text{oracle}}\}$ available in the simulator to construct a posterior latent distribution z_t, which guides the learning of the prior $\tilde{z}_t$. Applying Jensen's inequality, the log-likelihood of the conditional policy can be lower-bounded as:

$$\begin{aligned}\log \pi(a_t \mid o_t, h_t) \geq \mathbb{E}_{q_\psi(z_t \mid s^{\text{oracle}})}\big[\log p_\theta(a_t \mid o_t, h_t, \tilde{z}_t)\big] - \\ \text{KL}\big(q_\psi(z_t \mid s^{\text{oracle}}) \,\|\, p_\phi(\tilde{z}_t \mid o_t, h_t)\big)\end{aligned} \tag{8}$$

Summing over all time steps gives the variational lower bound (Evidence Lower Bound, ELBO) for a trajectory:

$$\begin{aligned}\mathcal{L}(\theta, \phi, \psi) = \sum_{t=0}^{T-1} \mathbb{E}_{q_\psi(z_t \mid s^{\text{oracle}})}\big[\log p_\theta(a_t \mid o_t, h_t, \tilde{z}_t)\big] - \\ \sum_{t=0}^{T-1} \text{KL}\big(q_\psi(z_t \mid s^{\text{oracle}}) \,\|\, p_\phi(\tilde{z}_t \mid o_t, h_t)\big)\end{aligned} \tag{9}$$

where θ, ϕ, ψ are the parameters of the decoder, prior network, and privileged encoder, respectively. This formulation shows that by introducing a prior network and performing variational inference with privileged states, the training procedure corresponds to a CVAE [28].

During inference, the privileged encoder is removed. The policy relies solely on the prior distribution $\tilde{z}_t$, and the decoder generates the action based on $(o_t, h_t, \tilde{z}_t)$. Since the prior network has been aligned with the posterior during training via the KL divergence, it implicitly models global action patterns. Consequently, the policy can generate diverse and temporally consistent action chunks even under partial observability. We note that although the slow policy trained with privileged information produces precise and stable action chunks in simulation, its inference is typically computationally expensive and open-loop, limiting its ability to react to dynamic environments in real time.

D. Tuning fast policy

In dynamic scenes, due to inference latency and the uncertainty of environmental responses, the action chunks generated by the slow policy $a_t \sim \pi(a_t \mid o_t, h_t)$ can probabilistically deviate from the ideal distribution. To enable rapid adaptation and closed-loop control, we propose the fast policy shown in Fig.2 (c) that models the executed action as a residual correction over the slow policy output:

$$\tilde{a}_t = a_t + \Delta a_t \tag{10}$$

where Δa_t represents an observation-dependent, instantaneous correction term. The residual is generated in two steps: first, a 1D convolutional network extracts basic action features from the slow policy output a_t and the previous residual Δa_{t-1}:

$$f_t = \text{1DCNN}(a_t, \Delta a_{t-1}) \tag{11}$$

which is then input to a lightweight RNN residual decoder g_ω along with recent observation history $o_{t-3:t}$ to produce the residual:

$$\Delta a_t = g_\omega(f_t, o_{t-3:t}) \tag{12}$$

Given the base action chunk a_t, the fast policy introduces a trainable residual mapping $g_\omega(f_t, o_{t-3:t})$, allowing fine-grained trajectory adjustments even under distributional shifts. The coupling between slow and fast policies can be expressed probabilistically as a joint policy distribution in closed form:

$$\begin{aligned}\pi_{\text{joint}}(\tilde{a}_t \mid o_t, h_t) = \int \delta\big(\tilde{a}_t - [a_t + g_\omega(f_t, o_{t-3:t})]\big) \\ \cdot\, p_\theta(a_t \mid o_t, h_t)\, da_t\end{aligned} \tag{13}$$

where $\delta(\cdot)$ is the Dirac delta function [29], enforcing a deterministic mapping between the residual-corrected action $\tilde{a}_t$ and the base action a_t. From a measure-theoretic perspective, π_{joint} corresponds to the pushforward measure of the prior distribution p_θ under the transformation $T_\omega(a_t) = a_t + g_\omega(\text{1DCNN}(a_t, \Delta a_{t-1}), o_{t-3:t})$.

During training, the fast policy is initialized from the slow policy checkpoint, while the parameters of the slow policy

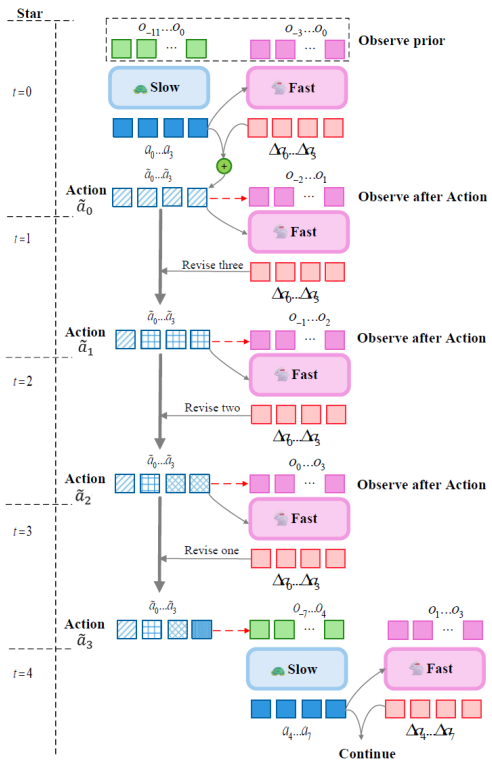

Fig. 3. **Temporal execution pipeline of the reactive action chunking with Slow-Fast policy.** At the initial step ($t = 0$), the slow policy generates a fundamental action chunk $a_0 \ldots a_3$ from long-term observations ($o_{-11} \ldots o_0$), while the fast policy produces refinement offsets $\Delta a_0 \ldots \Delta a_3$ from recent observations ($o_{-3} \ldots o_0$). The refined sequence $\tilde{a}_0 \ldots \tilde{a}_3$ is obtained by combining both outputs. At subsequent steps ($t = 1, 2, 3$), the fast policy iteratively revises the remaining unexecuted actions based on new observations, providing closed-loop feedback at a higher frequency. When reaching $t = 4$, the slow policy generates the next action chunk $a_4 \ldots a_7$, and the fast policy computes corresponding refinements $\Delta a_4 \ldots \Delta a_7$, repeating the process over time.

are frozen. Only the residual decoder g_ω is optimized via reinforcement learning. The formal objective is to maximize expected cumulative reward over trajectories:

$$\arg\max_{\omega} \mathbb{E}_{\tau \sim \pi_{\text{joint}}(\tilde{a}_t | o_t, h_t)} \left[\sum_t R(o_t, \tilde{a}_t) \right] \tag{14}$$

where τ denotes the state-action trajectory and R is the task-specific reward function [30]. As shown in Fig.3, in inference, the Slow-Fast policy works together to form a closed-loop system.The overall pipeline of REACT is summarized in Table I.

TABLE I
SLOW–FAST POLICY TRAINING

Algorithm: CVAE with Slow–Fast Policy
Slow-Policy (CVAE) Training:
1. Initialize oracle encoder p_ϕ, prior q_ψ, and action generator p_θ.
2. Sample observation o_t and action chunk $a_{t:t+k}$ from $\mathcal{D}$.
3. Sample latent $z \sim p_\phi(z \mid a_{t:t+k}, \bar{o}_t)$ and $z_p \sim q_\psi(z \mid \bar{o}_t)$.
4. Generate slow action chunk $\hat{a}^s_{t:t+k} = p_\theta(z \mid \bar{o}_t)$.
5. Compute KL divergence: $\mathcal{L}_{\text{KL}} = D_{\text{KL}}(p_\phi(z \mid a_{t:t+k}, \bar{o}_t) \,\|\, q_\psi(z \mid \bar{o}_t))$.
Fast-Policy Training:
6. latent feature: $h_t = \text{RNN}(a_t, \Delta a_{t-1}, \bar{o}_t)$.
7. Predict refined actions: $\hat{a}^f_{t:t+k} = \pi_\xi(h_t, \hat{a}^s_{t:t+k})$.
8. Compute reconstruction loss: $\mathcal{L}_{\text{L1}} = \|\hat{a}^s_{t:t+k} - a_{t:t+k}\|_1, \quad \mathcal{L}_{\text{MSE}} = \|\hat{a}^f_{t:t+k} - a_{t:t+k}\|_2^2$.
Optimization:
9. Compute total loss: $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{L1}} + \lambda_2 \mathcal{L}_{\text{MSE}} + \beta \mathcal{L}_{\text{KL}}$.
10. Update ϕ, ψ, θ, ξ using Adam optimizer.
11. Repeat steps 2–10 until convergence.

IV. EXPERIMENT

A. Simulation and reinforcement learning settings

The proposed data generation method was trained on the Isaac Lab platform using the RSL-RL framework. Policy optimization was performed using the Proximal Policy Optimization (PPO) [31] algorithm. During simulation, privileged information—specifically precise object coordinates and friction coefficients—is fully accessible. To enhance dataset diversity and model generalization, we implemented several Domain Randomization strategies: adjusting conveyor acceleration profiles, varying conveyor positions, and randomizing object dimensions and initial placements.

The reward function integrates multiple terms that jointly shape the grasping behavior, including object reaching, goal tracking, timing, and lifting. The dominant positive components (e.g., *Object Goal Tracking*, *Close Fingers*, *Lift Object*) encourage precise interaction, while penalties on prediction error and action rate promote stable execution.

During data collection, each rollout contains synchronized RGB observations, robot joint states (`qpos`), and action labels, recorded at 500 steps per trajectory. The full dataset consists of 52,000 trajectories, each represented as a sequence of 12-dimensional visual features and 14-dimensional joint configurations. A trajectory is considered successful if any step simultaneously achieves a reward greater than 0.35 and an orientation score exceeding 1.25.

B. Model training and sim-to-real method

All models were trained using two NVIDIA RTX 3090 GPUs, while inference was executed on a single RTX 4090 GPU. We employed a learning rate decay mechanism, starting with a higher initial rate that gradually decreases to ensure stability in later stages. Additionally, gradient clipping was utilized to stabilize the learning process of the high-frequency action adjustment mechanism.

The slow CVAE policy encodes action chunks with 4-/7-layer encoder/decoder and 512 hidden units, trained with L1 loss and KL regularization, while the fast RNN policy refines the plan using a 3-layer RNN and 512-unit actor MLP, both optimized with Adam.

For physical evaluation, we utilized the Cobot Magic platform. The robot is equipped with three Intel RealSense

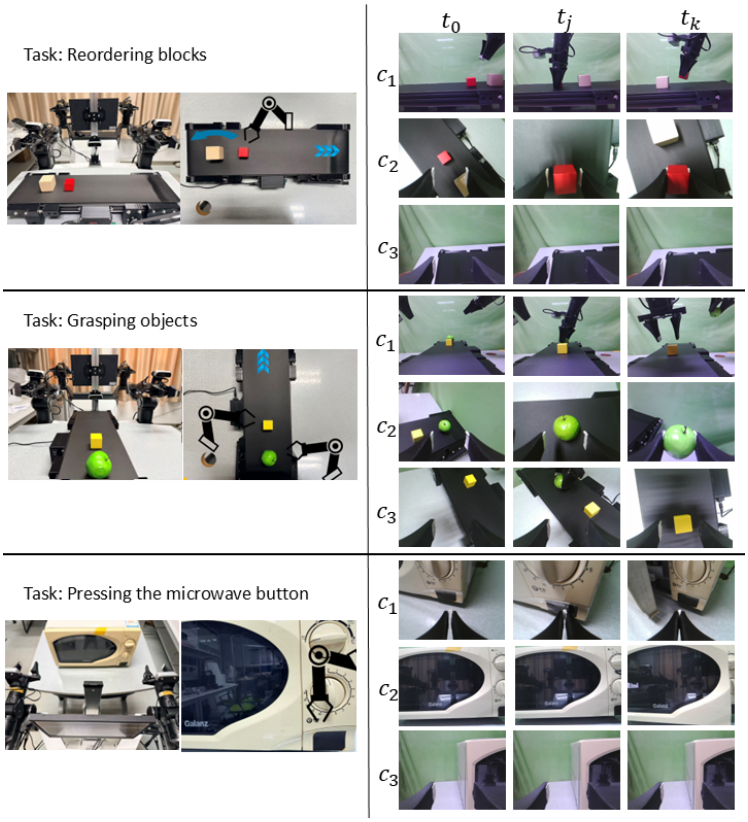

Fig. 4. **Inference visualization for three tasks.** C_1, C_2, and C_3 denote the camera views from the robot base, right arm, and left arm, respectively, captured at t_0, t_j, and t_k. **Task 1: Reordering blocks** The right arm moves a red blocks from one side of the conveyor to the other **Task 2: Grasping objects on Track.** The dual-arm agents alternately pick objects from the conveyor and place them at designated locations. **Task 3: Pressing the door-release button of a microwave.**

D435 cameras: two wrist-mounted cameras providing lateral perspectives and one torso-mounted camera for a frontal view. We selected three representative experimental scenarios (Fig. 4), The conveyor measures 80×20 cm and operates at a velocity of 2 cm/s. All tasks were conducted on a uniform gray tabletop.

To bridge the sim-to-real gap and address the sensor noise inherent in real-world deployment, we employed a Teacher-Student distillation framework inspired by DAgger [32]. Proposed architecture serves as the *Student Policy*, which only receives limited sensory inputs: raw RGB images and robot proprioception.

During the distillation phase, We iteratively update the Student Policy to minimize the reconstruction loss between its output and the Teacher's action. By incorporating Domain Randomization during this process, the Student Policy learns to infer the underlying physical states from noisy visual observations, ensuring robust performance during zero-shot transfer to the physical Cobot Magic platform.

C. Quantitative Analysis

1) Baseline Comparison: Most existing baselines focus on static tabletop manipulation. To quantitatively evaluate the performance of REACT, we conducted multiple manipulation tasks within the RoboTwin [33] environment. Table II summarizes the Success Rates (SR) for several tasks involving moving objects and dynamic manipulation. REACT was trained with a batch size of 16 across two GPUs for 6,000 epochs. Following the RoboTwin 2.0 standard protocol, each task was trained using 50 `demo_clean` demonstrations and evaluated over 100 simulation trials. Overall, REACT demonstrates performance that is comparable to or exceeds baseline models in static tabletop tasks.

TABLE II
COMPARISON WITH BASELINES ON ROBOTWIN TASKS

Tasks (easy)	act	DP3	RDT	REACT (ours)
Beat Block Hammer	56%	72%	**77%**	70%
Handover Block	42%	70%	45%	**76%**
Move Can Pot	22%	**70%**	25%	67%
Open Microwave	86%	61%	37%	**90%**

2) Real-world Results: Table III provides a detailed comparative analysis for the most representative task: Grasping Objects on Track. Each experiment was repeated 20 times to compare Imitating CNN [34], Transformer-DIL [35], ACT [10], DP3 [20], and our proposed REACT. Evaluation metrics include average Computation Time (CT), average Exe-

TABLE III
CONTRASTIVE RESULTS OF TASK: GRASPING OBJECTS ON TRACK

Method	CT(s)	ET(s)	TT(s)	RMSE	SR(%)
imitating CNN	2.2153	27.2786	29.4939	3.2132×10^{-1}	15%
Transformer-DIL	2.3718	26.1009	28.4727	2.9824×10^{-1}	35%
DP3	3.4211	28.5663	31.9874	1.9823×10^{-1}	60%
act	1.9831	**22.9262**	**24.9093**	1.5521×10^{-1}	45%
REACT (ours)	**1.2104**	24.2912	25.5016	**6.4511** $\times 10^{-2}$	**80%**

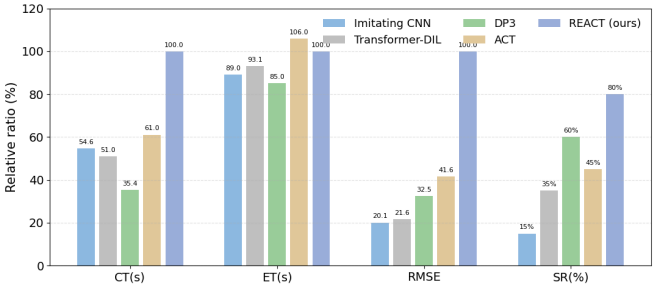

Fig. 5. **Comparison with the baseline.** Using the metrics of the REACT method as the normalization baseline, we present a visualized comparison between REACT and the baseline methods in terms of CT, ET, and RMSE; SR is displayed in its original scale.

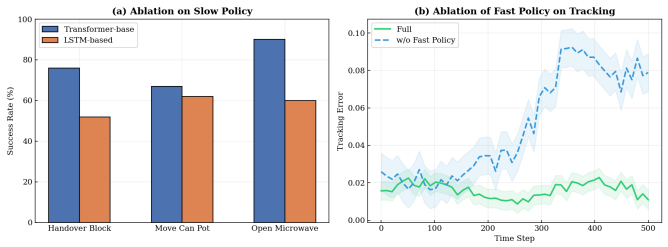

Fig. 6. **Ablation results.** (a) Transformer-based vs. LSTM-based CVAE for the slow policy. While both perform similarly on short-horizon tasks, the LSTM variant degrades sharply on long-horizon, multi-stage tasks. (b) Effect of the fast policy in dynamic settings. Without fast corrections, errors accumulate over time.

cution Time (ET), Total Time(TT)and Root Mean Square Error (RMSE) [36], completion times were calculated exclusively for successful executions, with values expressed as the mean, while Success Rate (SR) is expressed as a percentage.

As illustrated in Fig. 5, our method significantly reduces CT due to the integration of the "Fast" network. While the ET of REACT slightly trails the ACT method—attributed to the RNN fine-tuning which slows actions when nearing the target—REACT achieves superior RMSE and SR compared to the diffusion-based DP3 while maintaining lower CT and ET. This efficiency is credited to the Fast-Slow network architecture: the Slow policy provides stable, long-horizon planning for smooth global trajectories, while the Fast policy ensures rapid, low-latency adjustments in response to highly dynamic state changes.

D. Ablation Study

We conduct ablation studies to address two key questions:

1) Does Transformer structure provide sufficiently rich prior planning information?
2) Can using fast policy to form a closed-loop structure improve real-time performance and robustness?

To investigate the importance of the Transformer architecture in our slow policy for long-horizon trajectory planning, we conduct an ablation study by replacing the Transformer-based CVAE with an LSTM-based CVAE. In this variant, the sequential dependencies of the trajectory are modeled by Long Short-Term Memory units instead of the multi-head self-attention mechanism.

As shown in Fig.6 (a), while LSTM performs reasonably well in short-range tasks such as *Move Can Pot*, its success rate drops significantly in multi-stage tasks like *Open Microwave* and *Handover Block*. This degradation is primarily attributed to the inherent limitations of recurrent architectures in capturing extremely long-term dependencies. In contrast, our Transformer-based slow policy maintains a global receptive field, providing a more stable and coherent trajectory prior for the subsequent fast policy to refine.

Fig.6 (b) addresses the second research question. In dynamic evaluations, while the Slow Policy effectively plans the global trajectory, the absence of the Fast Policy's real-time adjustments leads to a rapid accumulation of temporal errors. This error peaks during the fine-manipulation phase as the end-effector approaches the target (at approximately step 300), validating the robust role of the Fast Policy in high-frequency, closed-loop error correction.

V. CONCLUSION

In this work, we revisit the limitations of existing open-loop and semi closed-loop paradigms in bimanual manipulation, which struggle to achieve real-time adaptation in dynamic scenes. Through large-scale simulated demonstrations, REACT establishes a closed-loop paradigm for bimanual manipulation based on Slow-Fast policy. We expect that this paper may offer a useful reference for future applications of bimanual agents in dynamic scenes. In future, we plan to improve the fast policy's motion refinement process, as sensor bias and latency can still cause minor discontinuities in motion smoothness.

REFERENCES

[1] J.-J. Jiang, X.-M. Wu, and et al, "Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025, pp. 12 427–12 437.

[2] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2410.07864

[3] M. Kobayashi, T. Buamanee, and T. Kobayashi, "ALPHA- α and bi-act are all you need: Importance of position and force information/ control for imitation learning of unimanual and bimanual robotic manipulation with low-cost system," *IEEE Access*, vol. 13, pp. 29 886–29 899, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3541200

[4] G. Lu, T. Yu, H. Deng, S. S. Chen, Y. Tang, and Z. Wang, "Anybimanual: Transferring unimanual policy for general bimanual manipulation," *arXiv preprint arXiv:2412.06779*, 2024.

[5] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[6] H. Tan, X. Xu, C. Ying, X. Mao, S. Liu, X. Zhang, H. Su, and J. Zhu, "Manibox: Enhancing spatial grasping generalization via scalable simulation data generation," 2024.

[7] S. Zhu, L. Mou, D. Li, B. Ye, R. Huang, and H. Zhao, "Vr-robo: A real-to-sim-to-real framework for visual robot navigation and locomotion," *IEEE Robotics Autom. Lett.*, vol. 10, no. 8, pp. 7875–7882, 2025. [Online]. Available: https://doi.org/10.1109/LRA.2025.3575648

[8] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[9] L. Lai, A. Z. Huang, and S. J. Gershman, "Action chunking as policy compression," *PsyArXiv*, 2022.

[10] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation," in *8th Annual Conference on Robot Learning*, 2024.

[11] Y. Wang, X. Yan, L. Wang, W. Pan, F. Song, and X. Zhou, "Design of imitation learning fusion algorithm for mobile robotic arm control," *Applied Soft Computing*, p. 113628, 2025.

[12] F. Xie, A. Chowdhury, M. De Paolis Kaluza, L. Zhao, L. Wong, and R. Yu, "Deep imitation learning for bimanual robotic manipulation," *Advances in neural information processing systems*, vol. 33, pp. 2327–2337, 2020.

[13] T. Kobayashi, M. Kobayashi, T. Buamanee, and Y. Uranishi, "Bi-lat: Bilateral control-based imitation learning via natural language and action chunking with transformers," *CoRR*, vol. abs/2504.01301, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.01301

[14] T. Ma, J. Zhou, Z. Wang, R. Qiu, and J. Liang, "Contrastive imitation learning for language-guided multi-task robotic manipulation," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 2024, pp. 4651–4669. [Online]. Available: https://proceedings.mlr.press/v270/ma25a.html

[15] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023. [Online]. Available: https://doi.org/10.15607/RSS.2023.XIX.016

[16] K. Black, M. Y. Galliker, and S. Levine, "Real-time execution of action chunking flow policies," *CoRR*, vol. abs/2506.07339, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.07339

[17] K. Yin, W. Zeng, K. Fan, M. Dai, Z. Wang, Q. Zhang, Z. Tian, J. Wang, J. Pang, and W. Zhang, "Unitracker: Learning universal whole-body motion tracker for humanoid robots," *arXiv preprint arXiv:2507.07356*, 2025.

[18] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[19] K. Lee, Y. Seo, S. Lee, H. Lee, and J. Shin, "Context-aware dynamics model for generalization in model-based reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5757–5766.

[20] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[21] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," 2026. [Online]. Available: https://arxiv.org/abs/2410.24164

[22] S. Dasari and A. Gupta, "Transformers for one-shot visual imitation," in *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. J. Tomlin, Eds., vol. 155. PMLR, 2020, pp. 2071–2084. [Online]. Available: https://proceedings.mlr.press/v155/dasari21a.html

[23] Y. Liu, X. Luo, and S. Xie, "Intention-guided imitation learning methods under limited expert demonstration data," *Knowledge-Based Systems*, p. 114455, 2025.

[24] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[25] M. Kelly, C. Sidrane, K. R. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*. IEEE, 2019, pp. 8077–8083. [Online]. Available: https://doi.org/10.1109/ICRA.2019.8793698

[26] K. Menda, K. R. Driggs-Campbell, and M. J. Kochenderfer, "Ensembledagger: A bayesian approach to safe imitation learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*. IEEE, 2019, pp. 5041–5048. [Online]. Available: https://doi.org/10.1109/IROS40897.2019.8968287

[27] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto, "Behavior generation with latent actions," *arXiv preprint arXiv:2403.03181*, 2024.

[28] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, and et al, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl

[29] R. Hartley, *Multiple view geometry in computer vision*. Cambridge university press, 2003, vol. 665.

[30] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu, "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," 2025, accepted for publication.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[32] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[33] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Q. Liang, X. Li, Zixuan andLin, Y. Ge, Z. 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*, 2025.

[34] S. Zhang, S. Li, Y. Li, X. Li, and Z. Wang, "A visual imitation learning algorithm for the selection of robots' grasping points," *Robotics and Autonomous Systems*, vol. 172, p. 104600, 2024.

[35] H. Kim, Y. Ohmura, and Y. Kuniyoshi, "Transformer-based deep imitation learning for dual-arm robot manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8965–8972.

[36] Y. Ning, T. Li, C. Yao, W. Du, Y. Zhang, and Y. Zhang, "Ts-ril: A two-stage robot imitation learning framework with motion trajectory learning and obstacle avoidance in real-world operating scenarios," *Robotics and Computer-Integrated Manufacturing*, vol. 97, p. 103111, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584525001656