AttenA+: Rectifying Action Inequality in Robotic Foundation Models

Daojie Peng[1]* **Fulong Ma**[1]* **Jiahang Cao**[2]* **Qiang Zhang**[1,3,6] **Xupeng Xie**[1]
Jian Guo[4] **Ping Luo**[2] **Andrew F. Luo**[2] **Boyu Zhou**[5] **Jun Ma**[1]†
[1]HKUST(GZ) [2]HKU [3]USTC
[4]IDEA Research [5]SUSTech [6]X-Humaniod

Abstract

Existing robotic foundation models, while powerful, are predicated on an implicit assumption of *temporal homogeneity*: treating all actions as equally informative during optimization. This "flat" training paradigm, inherited from language modeling, remains indifferent to the underlying physical hierarchy of manipulation. In reality, robot trajectories are fundamentally heterogeneous, where low-velocity segments often dictate task success through precision-demanding interactions, while high-velocity motions serve as error-tolerant transitions. Such a misalignment between uniform loss weighting and physical criticality fundamentally limits the performance of current Vision-Language-Action (VLA) models and World-Action Models (WAM) in complex, long-horizon tasks. To rectify this, we introduce **AttenA+**, an architecture-agnostic framework that prioritizes kinematically critical segments via velocity-driven action attention. By reweighting the training objective based on the inverse velocity field, AttenA+ naturally aligns the model's learning capacity with the physical demands of manipulation. As a plug-and-play enhancement, AttenA+ can be integrated into existing backbones *without structural modifications or additional parameters.* Extensive experiments demonstrate that AttenA+ significantly elevates the ceilings of current state-of-the-art models. Specifically, it improves OpenVLA-OFT to 98.6% (+1.5%) on the LIBERO benchmark and pushes FastWAM to 92.4% (+0.6%) on RoboTwin 2.0. Real-world validation on a Franka manipulator further showcases its robustness and cross-task generalization. Our work suggests that mining the intrinsic structural priors of action sequences offers a highly efficient, physics-aware complement to standard scaling laws, paving a new path for general-purpose robotic control. The code is available at: https://github.com/DaojiePENG/AttenA-Plus.

1 Introduction

Vision-Language-Action (VLA) and World-Action Models (WAM) have recently emerged as a powerful paradigm for end-to-end robotic control, enabling robots to interpret multimodal instructions and execute complex physical tasks [1, 2, 3]. However, this success masks a fundamental misalignment: while all linguistic tokens are assumed to be equally informative in standard NLP training, robotic actions are *inherently heterogeneous* in their physical significance.

In a typical manipulation trajectory, not all action steps are created equal. Consider the task of picking up a fragile object: the rapid motion of the arm toward the object is transitional and error-tolerant, whereas the final, slow-speed adjustment of the gripper is precision-demanding and task-critical. Currently, dominant training frameworks adopt a "flat" optimization objective,

*Equal contribution.
†Corresponding author: jun.ma@ust.hk

Preprint. Under review.

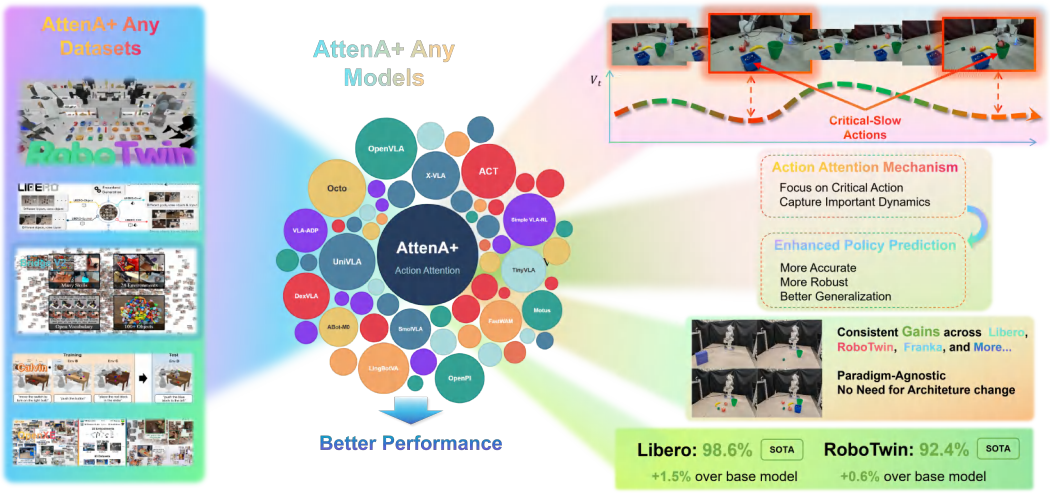

Figure 1: **Overview of AttenA+.** AttenA+ is a paradigm-agnostic enhancement framework for action robotic foundation models, introducing velocity-field-based action attention to prioritize slow, critical manipulation steps. It seamlessly plugs into mainstream discriminative (e.g., OpenVLA-OFT) and generative (π_0, $\pi_{0.5}$, Diffusion Policy) architectures, as well as emerging World-Action Models (WAM). Without modifying core backbones or relying on data/model scaling, AttenA+ generalizes across diverse robotic datasets including LIBERO [15] and RoboTwin [16], and consistently improves task success rates over state-of-the-art baselines.

assigning *identical learning weights* to every timestep regardless of its physical role [4, 5, 6]. This uniform treatment forces models to waste representational capacity on trivial transitional segments, while under-optimizing the slow, high-precision actions that actually determine task success [1, 7, 2, 8, 9, 10, 11]. Consequently, even the most advanced VLA models often struggle with last-centimeter precision in complex robotics tasks [12, 13, 14].

To bridge this gap, we argue that the physical properties of an action, specifically its velocity, should dictate its importance during training. We propose **AttenA+**, a universal framework that introduces *velocity-driven action attention* to reweight trajectory learning. Our key insight is simple yet effective: the end-effector's velocity serves as a natural inverse proxy for precision demand. By assigning different optimization priorities to different actions, AttenA+ aligns model training with the intrinsic physics of manipulation. As an architecture-agnostic enhancement, AttenA+ can be seamlessly plugged into any existing robotic backbone without structural modifications or additional parameters.

Our contributions are summarized as follows: **1)** We identify and formalize the *action inequality* inherent in robotic trajectories, exposing a fundamental bias in current foundation models where the uniform treatment of all actions leads to suboptimal optimization of physically critical steps. **2)** We introduce **AttenA+**, a plug-and-play optimization framework that utilizes the inverse velocity field as a physical prior to reweight trajectory learning, effectively aligning the model's focus with kinematically demanding manipulation phases. **3)** Extensive evaluations on LIBERO and RoboTwin benchmarks show that AttenA+ significantly elevates the performance ceilings of current state-of-the-art models. Furthermore, real-world experiments on a Franka manipulator demonstrate that our method provides superior robustness and success rates specifically during precision-critical motions where standard baselines frequently fail.

2 Related Works

2.1 Robotic Foundation Models

Vision-Language-Action (VLA) and World-Action Models (WAM) enable end-to-end robotic manipulation by grounding language in visual observations to generate continuous action sequences. A wide range of VLA frameworks have been proposed to advance robotic manipulation performance,

including foundational models and their variants, as well as specialized optimizations. OpenVLA [1] serves as a core foundational framework unifying visual perception, language understanding, and action generation, with its variant OpenVLA-OFT [17] further optimizing via orthogonal fine-tuning to push state-of-the-art (SOTA) performance on LIBERO tasks. The π model series, including π_0 [2], π_0 + FAST [18], and $\pi_{0.5}$ [10], advances generative VLA capabilities through flow matching for strong generalization. Other representative VLA models and optimizations include UniVLA [7], VLA-ADP [19], CogACT [20], SmolVLA [21], NORA and NORA-Long [22], WorldVLA and WorldVLA* [8], SP-VLA [23], FlashVLA [24], VLA-Cache [25], FastV and FastV(+OFT) [26], SparseVLM [27], and CSP [28]. Parallel efforts emerging as WAMs include Motus [13], LingBot-VA [14], and Fast-WAM [29]. Despite consistent progress across benchmarks, nearly all existing action models share a core limitation: treating all action timesteps equally during training, neglecting the intrinsic physical hierarchy and heterogeneous importance of different motion phases.

2.2 Action Sequence Modeling for Robotics

Modeling sequential robotic actions is a core research direction, with early efforts focusing on trajectory optimization and inverse reinforcement learning (IRL). Recent data-driven approaches include Action Chunking with Transformers (ACT) [30], which uses transformers to model temporal dependencies in action sequences, and Diffusion Policy [31], which leverages diffusion models for smooth, feasible trajectory generation—though these prioritize action quality over critical action prioritization based on physical characteristics (e.g., velocity). Prior works have explored importance weighting for imitation learning: some weight entire trajectories by demonstration quality [32, 33], while others focus on per-timestep weighting (e.g., IRIS [34] for informative offline data, uncertainty-based weighting for critical states [35]). However, these are limited to single-task learning or require extra overhead, unlike our velocity-based approach that needs no additional training and is compatible with arbitrary VLA frameworks. A small number of VLA works explore action weighting (e.g., VLA-ADP [19] prunes redundant fast actions for efficiency), but none explicitly link velocity to learning priority. Different from all prior arts, **AttenA+** introduces a plug-and-play velocity-field weighting principle that emphasizes learning on critical action phases, requiring no extra supervision and universally compatible with mainstream robotic foundation models.

2.3 Attention Mechanisms in Robotic Learning

Attention has become a standard component in modern robotic foundation models, yet its usage remains largely confined to input modalities. Visual attention focuses on task-relevant spatial regions for manipulation [36]; language attention aligns linguistic instructions with visual observations [37]; cross-modal attention further fuses vision and language features to predict actionable policies [1, 7]. Remarkably, almost all prior attention designs operate exclusively on input vision-language streams, while output action trajectories are treated as plain unweighted regression targets. Our work breaks this convention by proposing *action attention*: we apply attention weighting directly to output action sequences, guided by physical velocity priors to emphasize precision-demanding motion segments. This extends attention design from input feature alignment to physical-aware action trajectory modeling, mirroring the hierarchical nature of human motor control.

3 Methodology

3.1 The Homogeneity Bias in Robot Learning

Current robotic foundation models, regardless of their underlying architectures, typically formulate expert trajectory as a sequence modeled via either independent single-step forecasting or autoregressive generation, with uniform optimization weight across all time steps. This *uniform weighting* strategy is prevalent across both discriminative and generative paradigms. Formally, given a dataset $\mathcal{D}$ of expert trajectories, the general optimization objective can be expressed as:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\tau\sim\mathcal{D}} \left[\sum_{t=1}^{T} \mathcal{L}_t(\pi_\theta(s_t), \boldsymbol{a}_t) \right], \tag{1}$$

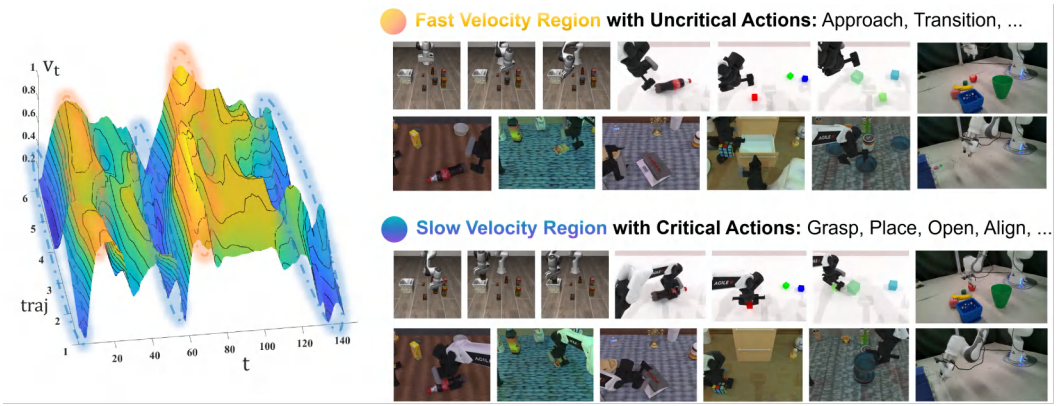

Figure 2: **Analysis of velocity fields revealing the inherent action inequality.** We observe that the informational density of the robot dataset is **non-uniformly distributed**: rapid motions are often redundant transitions, while slow-motion phases dominate task success or failure. The discovery of this kinematic hierarchy motivates the development of AttenA+, a plug-and-play mechanism designed to rectify the uniform weighting bias in current robotic foundation models.

where $\mathcal{L}_t$ is the per-step loss function. In discriminative models [1, 7], $\mathcal{L}_t$ often takes the form of a regression loss (e.g., L_1 or L_2); in generative models such as diffusion policy [31, 38, 39] or flow matching models (π_0 [2]), it corresponds to a score-matching or vector-field objective.

Despite the diversity in loss formulations, these paradigms share an implicit assumption of *temporal homogeneity*: every action token a_t contributes identically to the overall gradient. However, this assumption is physically misaligned with the reality of robotic manipulation. By reducing complex physical interactions to a flat sequence of undifferentiated control signals, existing models inadvertently waste representational capacity on redundant transitional motions while under-optimizing the high-stakes, precision-demanding segments that truly govern task success.

3.2 Quantifying Action Inequality via Velocity Fields

To rectify this misalignment, we propose a shift from uniform optimization toward *Kinematic Criticality*. Our approach is rooted in the empirical discovery of *Action Inequality*: the observation that the informational density of a manipulation sequence is non-uniformly distributed and is intrinsically linked to the movement velocity.

As visualized in Figure 2, we analyze the velocity distribution across diverse task datasets. The results reveal a clear physical hierarchy within action sequences. High-velocity regions (highlighted in warm colors) typically correspond to "approach" or "transitional" phases—motions that occur in free space and are highly error-tolerant. In contrast, low-velocity regions (cold colors) consistently align with "interaction-rich" phases, such as precise alignment, grasping, or delicate placement. In these slow-motion segments, even a minor prediction error ϵ can lead to catastrophic task failure due to tight environmental constraints or contact dynamics.

We formalize this relationship by defining the instantaneous velocity magnitude v_t of the ground-truth action $\boldsymbol{a}_t^{gt}$ at each timestep:

$$v_t = \|\boldsymbol{a}_t^{gt}\|_2 = \sqrt{\sum_{d=1}^{D_{pos}} (a_{t,d}^{gt})^2}, \tag{2}$$

where D_{pos} denotes the translational and/or rotational degrees of freedom. This metric v_t serves as a natural, unsupervised proxy for task importance: lower velocity signifies higher precision demand. This discovery motivates a re-weighting of the optimization landscape to prioritize these low-velocity, high-criticality actions. In the following section, we introduce the **AttenA+** framework, which leverages this velocity-based prior to adaptively rescale the loss contribution of individual action tokens across different learning paradigms.

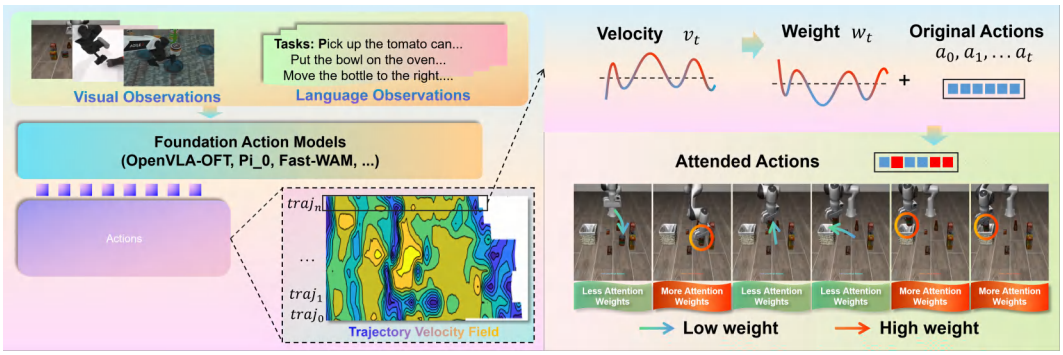

Figure 3: **Overview of AttenA+.** Given visual and language observations from datasets, we derive a velocity field. With attention weighting function F_A, this field assigns higher attention weights to slow, critical manipulation steps and lower weights to fast transitional motions, prioritizing learning on error-sensitive actions while training the models.

3.3 Velocity-Field Attention (AttenA+)

To rectify the uniform weighting bias identified in current paradigms, we introduce **AttenA+**, a velocity-aware weighting mechanism designed to align model optimization with the physical criticality of robotic manipulation. As illustrated in Figure 3, AttenA+ functions as a "plug-and-play" enhancer that re-scales the loss manifold across diverse learning objectives without requiring architectural modifications.

3.3.1 Weight Construction and Mapping

The core of AttenA+ lies in translating the kinematic properties of expert demonstrations into an optimization priority. For a given dataset $\mathcal{D}$, we derive the instantaneous velocity magnitude v_t following Equation 2. Taking the LIBERO benchmark ($D = 7$) as a representative case, we compute the velocity magnitude using the first 6 dimensions (joint velocities) of the ground-truth action sequence $\mathcal{A}^{gt}$, omitting the binary gripper state to focus on continuous motion dynamics. This approach ensures that the resulting weight matrix $\mathcal{W} \in \mathbb{R}^{T\times 1}$ captures the intrinsic difficulty of the maneuver: low-speed segments, which consistently align with task-critical phases such as object grasping or precision placement, are assigned higher learning priorities, while high-speed transitional movements are downweighted.

We define the attention weighting function F_A to map velocity to its corresponding importance weight:

$$w_t = F_A(v_t). \tag{3}$$

To accommodate varying task dynamics and noise profiles, we design four configurable mapping strategies: **inverse**, **inverse squared**, **exponential decay**, and **logarithmic**. These functions provide varying degrees of non-linear amplification for low-velocity actions, with detailed mathematical formulations provided in Appendix C.

3.3.2 Regularization for Training Stability

Directly applying raw inverse velocity weights can lead to numerical instability or gradient dominance by near-static timesteps. To ensure robust convergence, AttenA+ incorporates two essential regularization steps:

- **Weight Clipping**: We constrain the weights to a predefined range $[1/\text{clip}_{\max}, \text{clip}_{\max}]$. This prevents individual precision-critical steps from overwhelming the overall gradient and mitigates the impact of potential noise in expert demonstrations.
- **Loss Normalization**: We optionally normalize the weight vector such that $\frac{1}{T}\sum_{t=1}^{T} w_t \simeq 1$. This ensures that the global learning rate remains consistent with standard unweighted baselines, facilitating stable integration into existing training pipelines.

3.3.3 Paradigm-Agnostic Optimization Objectives

A defining advantage of AttenA+ is its **paradigm agnosticism**. It can be seamlessly integrated into diverse action models by augmenting the existing loss function.

Discriminative Models (AttenA+Disc): For standard regression-based VLAs, we transform the vanilla objective into a velocity-weighted L_1 loss:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(\mathcal{I},L,\mathcal{A}^{gt})\sim\mathcal{D}} \left[\frac{1}{T \cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} w_t \cdot |a_{t,d}^{\text{pred}} - a_{t,d}^{gt}| \right], \tag{4}$$

where θ denotes the model parameters and w_t represents the velocity-derived weight.

Flow Matching Models (AttenA+FM): For generative frameworks such as π_0 or $\pi_{0.5}$, we revise the flow-matching objective to guide the model toward learning more accurate flow fields specifically for high-criticality segments:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{\substack{(\mathcal{I},L,\mathcal{A}^{gt})\sim\mathcal{D} \\ \epsilon\sim\mathcal{N}(0,I)}} \left[\frac{1}{T \cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} w_t \cdot \|u_t(\epsilon;\mathcal{I},L) - (a_{t,d}^{gt} - \epsilon_d)\|_2^2 \right], \tag{5}$$

where u_t is the predicted flow field. By prioritizing these segments, AttenA+ enables generative models to capture the subtle nuances of precision-demanding actions that are often "washed out" in uniform training paradigms.

4 Experiment

We evaluate AttenA+ using four metrics: (1) Success Rate (SR) (%): percentage of successfully completed tasks. (2) Average Success Rate ($\overline{\text{SR}}$) (%): mean success rate across tasks. (3) Average Error Rate ($\overline{\text{ER}}$) (%): mean error rate across tasks. (4) Average Success Rate Improvement ($\overline{\text{SR-I}}$) (%): absolute gain in average success rate. (5) Average Relative Error Rate Reduction ($\overline{\text{RER-R}}$) (%): relative error reduction computed by

$$\overline{\text{RER-R}} = \left(1 - \frac{\overline{\text{ER}}_{\text{AttenA+}}}{\overline{\text{ER}}_{\text{other}}}\right) \times 100. \tag{6}$$

4.1 LIBERO and RoboTwin 2.0 Benchmark

We build AttenA+OFT upon the official OpenVLA-OFT framework, and benchmark our approach on LIBERO dataset (Figure 5-I-(a)) against representative state-of-the-art VLA and WAM models across all four task subsets of the LIBERO dataset. We select the best-performing checkpoint from training, then conduct evaluation across 4 random seeds to report the mean and standard deviation of success rates. Additional training configurations are provided in Appendix E.1. As summarized in Table 1, AttenA+OFT obtains an overall average success rate of 98.6%, surpassing the prior SOTA OpenVLA-OFT by 1.5%. Consistent performance gains are observed across all task categories. In particular, our method achieves a 2.1% improvement on long-horizon manipulation tasks, verifying that our action attention mechanism effectively enhances robustness and precision for complex, extended sequential behaviors.

We further validate our method on the RoboTwin benchmark (Figure 5-I-(b)), implementing AttenA+WAM based on the Fast-WAM framework. As shown in Table 2, AttenA+WAM achieves a new state-of-the-art average success rate of 92.46%, improving the base model Fast-WAM by 0.6% and outperforming the prior best LingBot-VA by 0.3%, without requiring any embodied pre-training. This confirms that our action attention mechanism can effectively boost performance even on larger, more diverse real-world benchmarks.

4.2 Improvement of Different Models with Action Attention

As shown in Table 3, we validate the effectiveness and generality of our velocity-field-based action attention by integrating it into both discriminative and generative models. Figure 4 provides a qualitative comparison: the original baseline fails due to accumulated errors in slow, critical manipulation

Table 1: Performance on LIBERO Compared with SOTA Methods. $\overline{\text{SR}}$(%): Average Success Rate; $\overline{\text{ER}}$(%): Average Error Rate; $\overline{\text{SR-I}}$(%): Average Success Rate Improvement; $\overline{\text{RER-R}}$(%): Average Relative Error Rate Reduction (Compared with AttenA+OFT using Equation 6).

Method	**Spatial**	**Object**	**Goal**	**10**	**$\overline{\text{SR}}$ ↑**	**$\overline{\text{ER}}$ ↓**	**$\overline{\text{SR-I}}$**	**$\overline{\text{RER-R}}$**
OpenVLA [1]	84.7	88.4	79.2	53.7	76.50	23.50	+22.1	-94.0
SparseVLM [27]	79.8	67.0	72.6	39.4	64.70	35.30	+33.9	-96.0
FastV [26]	83.4	84.0	74.2	51.6	73.30	26.70	+25.3	-94.8
VLA-Cache [25]	83.8	85.8	76.4	52.8	74.70	25.30	+23.9	-94.5
FlashVLA [24]	84.2	86.4	75.4	51.4	74.35	25.65	+24.3	-94.5
SP-VLA [23]	75.4	85.6	84.4	54.2	74.90	25.10	+23.7	-94.4
WorldVLA [8]	85.6	89.0	82.6	59.0	79.05	20.95	+19.6	-93.3
NORA-Long [22]	92.2	95.4	89.4	74.6	87.90	12.10	+10.7	-88.4
SmolVLA [21]	93.0	94.0	91.0	77.0	88.75	11.25	+9.9	-87.6
CogACT [20]	97.2	98.0	90.2	88.8	93.55	6.45	+5.1	-78.3
CSP [28]	84.7	82.2	77.1	74.3	79.58	20.42	+19.1	-93.1
π_0 + FAST [18]	96.4	96.8	88.6	60.2	85.50	14.50	+13.1	-90.3
π_0 [2]	96.8	98.8	95.8	85.2	94.15	5.85	+4.6	-76.1
$\pi_{0.5}$ [10]	98.8	98.2	98.0	92.4	96.85	3.15	+1.8	-55.6
UniVLA [7]	96.5	96.8	95.6	92.0	95.23	4.77	+3.4	-70.7
VLA-ADP [19]	**99.0**	98.2	96.8	91.2	96.30	3.70	+2.3	-62.2
OpenVLA-OFT [17]	97.6	98.4	97.9	94.5	97.10	2.90	+1.5	-51.7
AttenA+OFT (ours)	**99.0 ± 0.16**	**100 ± 0.00**	**98.8 ± 0.28**	**96.6 ± 0.30**	**98.60**	**1.40**	-	-

Table 2: Performance on RoboTwin 2.0 Compared with SOTA Methods.

Method	Embodied PT.	Clean	Rand.	**$\overline{\text{SR}}$ ↑**	**$\overline{\text{ER}}$ ↓**	**$\overline{\text{SR-I}}$**	**$\overline{\text{RER-R}}$**
π_0 [2]	✓	65.92	58.40	62.20	37.80	+30.3	-80.1
$\pi_{0.5}$ [10]	✓	82.74	76.76	79.75	20.25	+12.7	-62.8
X-VLA [40]	✓	72.90	72.80	72.85	27.15	+19.6	-72.2
Motus [13]	✓	88.66	87.02	87.80	12.20	+4.6	-38.0
LingBot-VA [14]	✓	<u>92.90</u>	91.50	<u>92.2</u>	<u>7.80</u>	+0.3	-3.3
Fast-WAM [29]	✗	91.88	<u>91.78</u>	91.80	8.20	+0.6	-7.7
AttenA+WAM (ours)	✗	**93.06**	**91.86**	**92.46**	**7.54**	-	-

steps (clip, align, release), where precision is essential but receives equal loss weight to fast transitional motions. In contrast, AttenA+ prioritizes these high-precision segments with larger attention weights, leading to stable task completion.

For the discriminative framework, we apply our method to OpenVLA-OFT, a strong baseline already achieving high performance on the LIBERO benchmark. Equipped with action attention, AttenA+OFT yields consistent gains across all task categories: Spatial (+1.4%), Object (+1.6%), Goal (+0.9%), and Long-horizon tasks (+2.1%). The overall average success rate improves by +1.5% (from 97.1% to 98.6%), with a corresponding -1.5% reduction in error rate. For the generative framework, we adopt $\pi_{0.5}$ as the backbone and construct AttenA+$\pi_{0.5}$. Similarly, consistent improvements are observed across all task types, with an average success rate increase of +1.10%.

These results demonstrate that our velocity-field action attention is *paradigm-agnostic* and can serve as a universal plug-and-play enhancement for both discriminative and generative models. Notably, the performance gain is most pronounced on long-horizon tasks, where distinguishing critical actions from transitional movements is essential for maintaining execution success.

Table 3: Performance Improvement with Velocity-Field-Based Action Attention on LIBERO dataset.

Model		**Spatial**	**Object**	**Goal**	**10**	**$\overline{\text{SR}}$**	**$\overline{\text{ER}}$**
Generative	$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.85	3.15
	AttenA+$\pi_{0.5}$	99.2 (+0.4)	99.6 (+1.4)	98.8 (+0.8)	94.2 (+1.8)	97.95 (+1.10)	2.05 (-1.10)
Discriminative	OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1	2.9
	AttenA+OFT	99.0 (+1.4)	100 (+1.6)	98.8 (+0.9)	96.6 (+2.1)	98.6 (+1.50)	1.4 (-1.50)

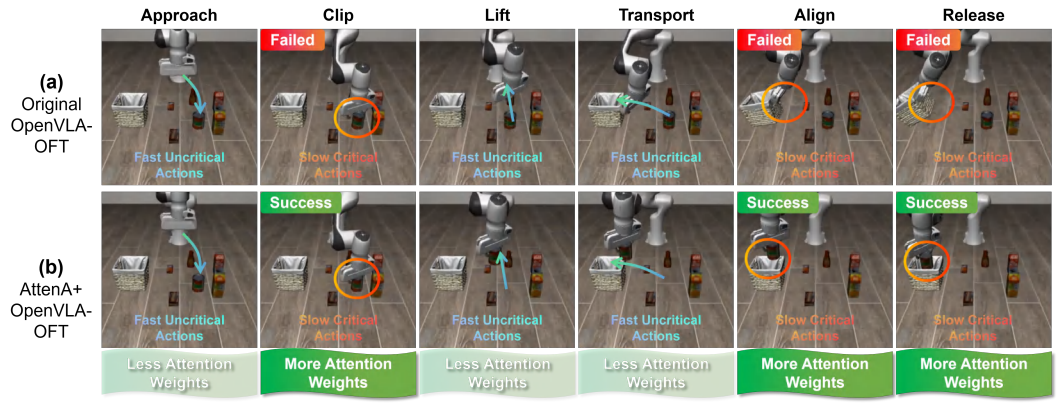

Figure 4: **Qualitative comparison of task execution with/without AttenA+.** (a) The original baseline fails due to accumulated errors in slow, critical manipulation steps (clip, align, release), which receive equal loss weight to fast transitional motions. (b) AttenA+ prioritizes these high-precision segments with larger attention weights, leading to stable task completion.

Table 4: Ablation study on different velocity weighting strategies and weight clipping thresholds $clip_{\max}$. We report task success rate (%) on the LIBERO benchmarks. Baseline is OpenVLA-OFT.

Strategy / Clip$_{\max}$	Libero-Spatial 2.0		Libero-Object 2.0		Libero-10 2.0		Libero-Goal 2.0		Libero-Goal 3.0		Libero-Goal 5.0	
	SR	ΔSR	SR	ΔSR	SR	ΔSR	SR	ΔSR	SR	ΔSR	SR	ΔSR
baseline	*97.6*	-	*98.4*	-	*94.5*	-	*97.9*	-	*97.9*	-	***97.9***	-
exp_decay ($w_{b,t} = e^{-\alpha \cdot v_{b,t}}$)	99.2	+1.6	99.8	+1.4	**96.8**	+2.3	**99.0**	+1.1	95.4	-2.5	97.4	-0.5
inverse_squared ($w_{b,t} = \frac{1}{v_{b,t}^2}$)	**99.4**	+1.8	99.8	+1.4	94.2	-0.3	98.8	+0.9	97.9	0.0	97.6	-0.3
inverse ($w_{b,t} = \frac{1}{v_{b,t}}$)	98.6	+1.0	**100.0**	+1.6	95.8	+1.3	98.0	+0.1	**98.2**	+0.3	95.6	-2.3
log ($w_{b,t} = \frac{1}{\log(1+v_{b,t})}$)	98.2	+0.6	99.6	+1.2	88.8	-5.7	**99.0**	+1.1	97.8	-0.1	97.8	-0.1

4.3 Ablation Study on Weighting Strategies and Clipping Thresholds

We conduct an ablation study to validate the effectiveness of our proposed velocity-based weighting strategies and the criticality of the weight clipping threshold clip$_{\max}$, with OpenVLA-OFT as our baseline model. Results on the LIBERO benchmark are reported in Table 4.

First, we observe that *no single weighting strategy universally dominates all task categories*, which aligns with the distinct motion characteristics of different robotic manipulation tasks. Specifically, `inverse_squared` achieves the best performance on LIBERO-SPATIAL, `inverse` performs optimally on LIBERO-OBJECT and clip$_{\max}$=3.0 settings of LIBERO-GOAL, while `exp_decay` and `log` show strong advantages on LIBERO-10 and clip$_{\max}$=2.0 settings of LIBERO-GOAL. This demonstrates that different velocity-aware weighting functions adapt to task-specific motion patterns.

Second, the clipping threshold clip$_{\max}$ plays a vital role in balancing weight emphasis and training stability. When clip$_{\max}$=1.0, all weighted loss terms degenerate to the uniform baseline, yielding identical performance to the original OpenVLA-OFT. As clip$_{\max}$ increases to 2.0 or 3.0, our AttenA+ mechanism consistently improves the task success rates. However, an overlarge threshold (clip$_{\max}$=5.0) tends to degrade performance, as extreme weights introduce training instability and over-emphasize noisy low-velocity actions. These results confirm that appropriate weight clipping is essential for maintaining the effectiveness of our velocity-field attention mechanism.

4.4 Real-World Robot Experiments

As shown in Figure 5, we design 4 kinds of task for validation using the Franka manipulator: *(a) Close the open drawer, (b) Put the Green Cube into Green Bowl, (c) Put Object-A into Green Bowl, (d) Put Object-A into XXX and then put Object-B into XXX.* For the easy tasks (a) and (b), we collect 50 trajectories for demonstration. For harder tasks (c) and (d), we collect 100 trajectories for

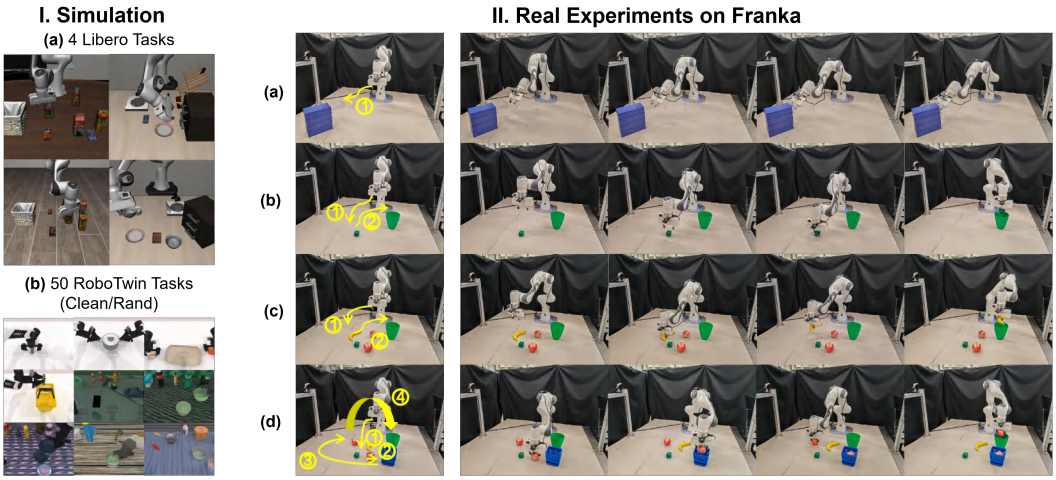

Figure 5: **Overview of experimental tasks.** I. Simulation: (a) Four LIBERO benchmark tasks; (b) 50 diverse RoboTwin tasks, including clean and randomized environments. II. Real-world experiments on Franka Panda: (a)–(d) Four representative tasks (drawer opening, pick-and-place, multi-objects, and sequential manipulation), showing AttenA+ enhanced policy execution.

Model	Close Draw	Put Cube	Multi-object	Long
OpenVLA-OFT	50/50	48/50	45/50	42/50
AttenA+OFT	50/50	50/50	49/50	45/50
Improvement	0%	4% ↑	8% ↑	6% ↑

Figure 6: **Real robot experimental results on Franka** (Each task is tested over 50 trials): (a) Quantitative success rates (%); (b) Qualitative performance visualization.

demonstration. Notably, during demonstration data collection, we use different speed for different phase: at the beginning, we use the baseline speed for approaching the object for grasping, then we change the speed to be 1/3 of the baseline to fine align and operate the object which indicates critical actions. Then after grasping the object we change the speed to baseline and fastly move to the bowl. When approaching the bowl, the speed is again reduced to 1/3 of the baseline for fine align to the bowl and finally release the object. After collection, we clean the trajectory by removing the no action waiting frames and do action smoothing for efficient training and action attention.

We then finetune and test the task following the OpenVLA-OFT recipe with 2 Nvidia H800 GPUs. In the testing phase, we deploy the model on a RTX-4090 GPU, evaluate each task for 50 times and compute the $\overline{\mathbf{SR}}$. The results are shown in Figure 6. We can see that AttenA+OFT consistently outperforms the baseline OpenVLA-OFT across all real-world tasks, improving the average success rate from 92.5% to 97.0%, with the largest gains on the more complex multi-object and long-horizon tasks, further validating the effectiveness of our method in real-world scenarios.

5 Conclusion

This work presents **AttenA+**, a generic enhancement framework for robotic foundation models. It introduces velocity-field-based action attention to prioritize critical, low-speed manipulation steps during training, aligning model optimization with real-world manipulation physics without modifying core architectures. Evaluated on LIBERO and RoboTwin 2.0 benchmarks, AttenA+ consistently improves success rates and reduces errors across both discriminative and generative paradigms (including VLA and WAM), and is readily extendable to other architectures such as diffusion policies.

We also note two main limitations. First, our velocity-weighted attention relies on a hand-crafted heuristic, which assumes critical manipulation steps are inherently slow. This does not generalize to dynamic tasks (e.g., high-speed grasping, table tennis) where critical actions may instead be fast and

ballistic. Second, the mechanism only leverages velocity information, ignoring other physical cues such as force or torque that can signal action importance.

Future work will move beyond fixed heuristics toward *physically grounded, learnable action attention* that integrates multi-modal physical signals and adapts dynamically to task semantics. By respecting the structure of robotic actions rather than treating all timesteps equally, we can build more efficient, robust, and generalizable robotic foundation systems.

References

[1] 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.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[3] Daojie Peng, Fulong Ma, and Jun Ma. Structured observation language for efficient and generalizable vision-language navigation. *arXiv preprint arXiv:2603.27577*, 2026.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

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

[6] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[7] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[8] 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.

[9] Daojie Peng, Jiahang Cao, Qiang Zhang, and Jun Ma. Lovon: Legged open-vocabulary object navigator. *arXiv preprint arXiv:2507.06747*, 2025.

[10] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi05: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[11] Jiahang Cao, Yize Huang, Hanzhong Guo, Qiang Zhang, Rui Zhang, Weijian Mai, Mu Nan, Jiaxu Wang, Hao Cheng, Jingkai SUN, Gang Han, Wen Zhao, Yijie Guo, Qihao Zheng, Xiao Li, Chunfeng Song, Ping Luo, and Andrew Luo. Compose your policies! improving diffusion-based or flow-based robot policies via test-time distribution-level composition. In *The Fourteenth International Conference on Learning Representations*, 2026.

[12] Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. *arXiv preprint arXiv:2502.08623*, 2025.

[13] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

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

[15] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[16] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, 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*, 2025.

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

[18] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[19] Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. *arXiv preprint arXiv:2509.22093*, 2025.

[20] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

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

[22] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

[23] Ye Li, Yuan Meng, Zewen Sun, Kangye Ji, Chen Tang, Jiajun Fan, Xinzhu Ma, Shutao Xia, Zhi Wang, and Wenwu Zhu. Sp-vla: A joint model scheduling and token pruning approach for vla model acceleration. *arXiv preprint arXiv:2506.12723*, 2025.

[24] Xudong Tan, Yaoxin Yang, Peng Ye, Jialin Zheng, Bizhe Bai, Xinyi Wang, Jia Hao, and Tao Chen. Think twice, act once: Token-aware compression and action reuse for efficient inference in vision-language-action models. *arXiv preprint arXiv:2505.21200*, 2025.

[25] Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Efficient vision-language-action manipulation via adaptive token caching. *Advances in Neural Information Processing Systems*, 38:164448–164473, 2026.

[26] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[27] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

[28] Xiaohuan Pei, Tao Huang, and Chang Xu. Cross-self kv cache pruning for efficient vision-language inference. *arXiv preprint arXiv:2412.04652*, 2024.

[29] 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.

[30] 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.

[31] 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.

[32] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[33] Voot Tangkaratt, Nontawat Charoenphakdee, and Masashi Sugiyama. Robust imitation learning from noisy demonstrations. *arXiv preprint arXiv:2010.10181*, 2020.

[34] Ajay Mandlekar, Fabio Ramos, Byron Boots, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Dieter Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. *arXiv preprint arXiv:1911.05321*, 2019.

[35] Qing Wang, Jiechao Xiong, Lei Han, Han Liu, Tong Zhang, et al. Exponentially weighted imitation learning for batched historical data. *Advances in Neural Information Processing Systems*, 31, 2018.

[36] Yiqi Huang, Travis Davies, Jiahuan Yan, Jiankai Sun, Xiang Chen, and Luhui Hu. Spatial robograsp: Generalized robotic grasping control policy. *arXiv preprint arXiv:2505.20814*, 2025.

[37] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[38] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[39] Jiahang Cao, Qiang Zhang, Jingkai Sun, Jiaxu Wang, Hao Cheng, Yulin Li, Jun Ma, Kun Wu, Zhiyuan Xu, Yecheng Shao, et al. Mamba policy: Towards efficient 3d diffusion policy with hybrid selective state models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11359–11366. IEEE, 2025.

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

[41] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[42] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

A Preliminary Concepts

A.1 Unified Task Definition and Notation

We unify the formulation for robotic foundation models (including VLA and WAM) across discriminative and generative paradigms using consistent notation, focusing on the core mapping from multimodal inputs to task-compliant action sequences:

- $\mathcal{I} = \{i_1, i_2, ..., i_T\}$: Sequence of visual observations (RGB images) at timesteps $1 \leq t \leq T$, where $i_t \in \mathbb{R}^{H \times W \times 3}$ in most cases.
- L: Natural language instruction (e.g., "stack the blue block on the red block"), tokenized to $L = [l_1, l_2, ..., l_N]$ via standard tokenizers (e.g., CLIP [41], Bert [42]).
- $\mathcal{A} = \{a_1, a_2, ..., a_T\}$: Robotic action sequence, with $a_t \in \mathbb{R}^D$ and D denoting action dimensions (e.g., $D = 7$ for LIBERO benchmarks).
- $\mathcal{A}^{gt}$: Ground-truth action sequence from expert demonstrations, serving as the target for model learning.

A.2 Discriminative Robotic Foundation Models

Discriminative paradigms [1, 17, 7] cast robotic action learning as a deterministic regression task. Given visual observations $\mathcal{I}$ and language instructions L, a discriminative model f_θ (parameters θ) directly predicts a deterministic action sequence:

$$\mathcal{A}^{\text{pred}} = f_\theta(\mathcal{I}, L) \tag{7}$$

Training minimizes the discrepancy between predicted and ground-truth actions via a regression loss over the dataset $\mathcal{D}$ (tuples of $\mathcal{I}, L, \mathcal{A}^{\text{gt}}$). Most existing works adopt an unweighted ℓ_1 loss as the optimization objective:

$$\theta^* = \arg\min_\theta \mathbb{E}_{(\mathcal{I}, L, \mathcal{A}^{\text{gt}}) \sim \mathcal{D}} \left[\frac{1}{T \cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} \left| a_{t,d}^{\text{pred}} - a_{t,d}^{\text{gt}} \right| \right] \tag{8}$$

A.3 Generative Robotic Foundation Models via Flow Matching

The π-series models (π_0 [2], $\pi_{0.5}$ [10]) are representative generative frameworks based on flow matching. Instead of direct regression, these models learn a continuous vector field to transform Gaussian random noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into task-aligned action sequences. Specifically, the learnable network g_ϕ predicts a time-dependent flow field conditioned on $\mathcal{I}$ and L, denoising random inputs toward $\mathcal{A}^{\text{gt}}$. The standard unweighted flow matching objective is:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{\substack{(\mathcal{I},L,\mathcal{A}^{\text{gt}})\sim\mathcal{D} \\ \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\frac{1}{T\cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} \left\| u_t(\epsilon; \mathcal{I}, L) - \left(a_{t,d}^{\text{gt}} - \epsilon_d\right) \right\|_2^2 \right] \tag{9}$$

Here, $a_{t,d}^{\text{gt}}$ is the d-th dimension of $\mathcal{A}^{\text{gt}}$ at timestep t, with T and D denoting action sequence length and dimensionality, respectively. While $\pi_{0.5}$ introduces hierarchical reasoning for complex tasks, it retains the original flow matching objective. Related diffusion policy frameworks are discussed in Appendix B.

B Action Attention Formulation of Generative VLA: Diffusion Policy

B.1 Generative VLA: Diffusion Model (Diffusion Policy)

Diffusion Policy (DP) represents an independent line of generative work, distinct from π_0/$\pi_{0.5}$. It models action generation as a iterative denoising process. Given K diffusion steps, the model h_ψ predicts the noise ϵ_k^{pred} added to the action state at step k. The standard unweighted L_2 diffusion optimization objective is:

$$\psi^* = \arg\min_{\psi} \mathbb{E}_{\substack{(\mathcal{I},L,\mathcal{A}^{gt})\sim\mathcal{D} \\ k\sim\text{Uniform}(1,K) \\ \epsilon_k\sim\mathcal{N}(0,I)}} \left[\frac{1}{T\cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} \left\| \epsilon_k^{\text{pred}} - \epsilon_k \right\|_2^2 \right] \tag{10}$$

where $a_{t,d}^{(k)} = \alpha_k a_{t,d}^{gt} + \beta_k \epsilon_k$ is the noisy action, and α_k, β_k are pre-defined diffusion schedule coefficients.

B.2 Revised Optimization Objective for Diffusion Policy (AttenA+Diff)

For diffusion-based generative policies (Diffusion Policy), we apply the same velocity-based weighting to the denoising objective:

$$\psi^* = \arg\min_{\psi} \mathbb{E}_{\substack{(\mathcal{I},L,\mathcal{A}^{gt})\sim\mathcal{D} \\ k\sim\text{Uniform}(1,K) \\ \epsilon_k\sim\mathcal{N}(0,I)}} \left[\frac{1}{T\cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} w_t \cdot \left\| \epsilon_k^{\text{pred}} - \epsilon_k \right\|_2^2 \right] \tag{11}$$

where w_t emphasizes denoising accuracy for slow, critical timesteps during the diffusion process.

C Velocity-Based Action Attention Weighting Strategies

We detail the four handcrafted weighting strategies used to implement velocity-field-based action attention in AttenA+. These formulations serve as *empirical, physics-inspired examples* to demonstrate the core idea of prioritizing slow, critical actions during training, rather than definitive or exclusive solutions. All weighting rules assign higher importance to low-velocity timesteps, consistent with the intuition that precise manipulation phases require stricter optimization in our experiment datasets (LIBERO, RoboTwin 2.0).

For the b-th sample at timestep t, the velocity-aware weight $w_{b,t}$ is defined as follows:

1. **Inverse strategy**

$$w_{b,t} = \frac{1}{v_{b,t}} \tag{12}$$

This baseline scheme applies inverse weighting proportional to action speed, providing a mild but clear emphasis on slower movements.

2. **Inverse squared strategy (amplified weight difference)**

$$w_{b,t} = \frac{1}{v_{b,t}^2} \tag{13}$$

By squaring the velocity term, this strategy strongly amplifies the contrast between slow and fast actions, making it the default choice in our main experiments.

3. **Exponential decay strategy (fast attenuation)**

$$w_{b,t} = e^{-\alpha \cdot v_{b,t}} \tag{14}$$

where $\alpha = 5.0$ controls the decay rate. This method suppresses high-speed actions rapidly while maintaining soft weighting for slow segments.

4. **Logarithmic strategy (smoothed weight)**

$$w_{b,t} = \frac{1}{\log(1 + v_{b,t})} \tag{15}$$

The logarithmic transform yields gentle, stable weighting, reducing sensitivity to noise in velocity estimation.

Notably, these four heuristic functions are *example implementations* chosen for simplicity, interpretability, and empirical effectiveness. They are not intended to limit the design space of action attention. In future work, action weighting can be naturally extended to broader families of parametric functions, task-adaptive formulations, or *fully learnable attention mechanisms* that infer importance end-to-end from data and physical constraints, rather than relying on fixed handcrafted rules.

D Visualization of Action Speed and Velocity-Guided Action Attention

In this section, we provide detailed analysis of the action speed patterns and velocity-guided attention weights, which are briefly summarized in the main text.

Figures 7–10 present comprehensive visualizations of raw action velocity profiles and the resulting velocity-field-based attention weights under four distinct clipping thresholds $clip_{\max} \in \{1.0, 2.0, 5.0, 10.0\}$ on the LIBERO-OBJECT manipulation benchmark. Each figure follows a consistent layout: Subplot 1 illustrates the temporal distribution of raw action speed magnitudes across multiple expert demonstration trajectories, revealing inherent speed variations within task execution. Subplots 2 through 5 display the attention weight distributions generated by the four velocity transformation rules (Equations 12–15): inverse weighting, inverse squared weighting, exponential decay weighting, and logarithmic weighting, respectively. Subplot 6 serves as the baseline, showing the original action trajectory under uniform, unweighted treatment without action attention.

From the raw velocity visualizations, distinct slow–fast motion patterns emerge consistently across all task trajectories. Slow-motion segments consistently align with task-critical phases, including robot initialization, precise object approaching, fine manipulation, grasping, targeted placement, and task completion. These phases demand high positional accuracy and are highly sensitive to execution errors, making them decisive for overall task success. Conversely, fast-motion segments correspond to robust, error-tolerant transitional movements, such as free-space arm traversal, coarse positioning toward target objects, and post-grasp repositioning, where minor deviations rarely lead to task failure.

The effect of the clipping threshold $clip_{\max}$ is clearly demonstrated across Figures 7–10. At $clip_{\max} = 1.0$, all velocity-adaptive weighting schemes collapse to uniform values, reverting AttenA+ to a standard VLA model with equal emphasis on all timesteps. As $clip_{\max}$ increases sequentially from 1.0 to 2.0, 5.0, and 10.0, the discriminative power of action attention is progressively strengthened: slow critical actions receive increasingly prominent weights, while fast transitional actions are assigned relatively lower weights, widening the gap in learning priority.

Moreover, the four velocity mapping functions exhibit distinctive attention characteristics. As clearly visible in the $clip_{\max} = 2.0$ visualization, exponential decay weighting (Equation 14) produces highly localized emphasis: it strongly amplifies a small set of extremely slow actions while broadly suppressing fast actions across a wide range. In contrast, inverse (Equation 12), inverse squared (Equation 13), and logarithmic (Equation 15) schemes maintain widespread emphasis on slow

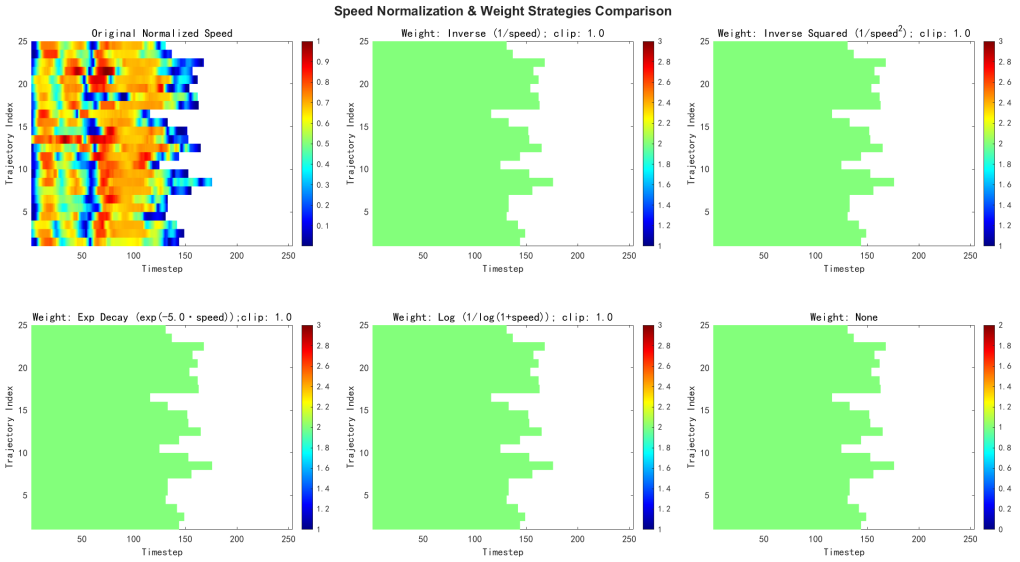

Figure 7: Visualization of Action Speed in LIBERO-OBJECT Task with Different $clip_{max} = 1.0$

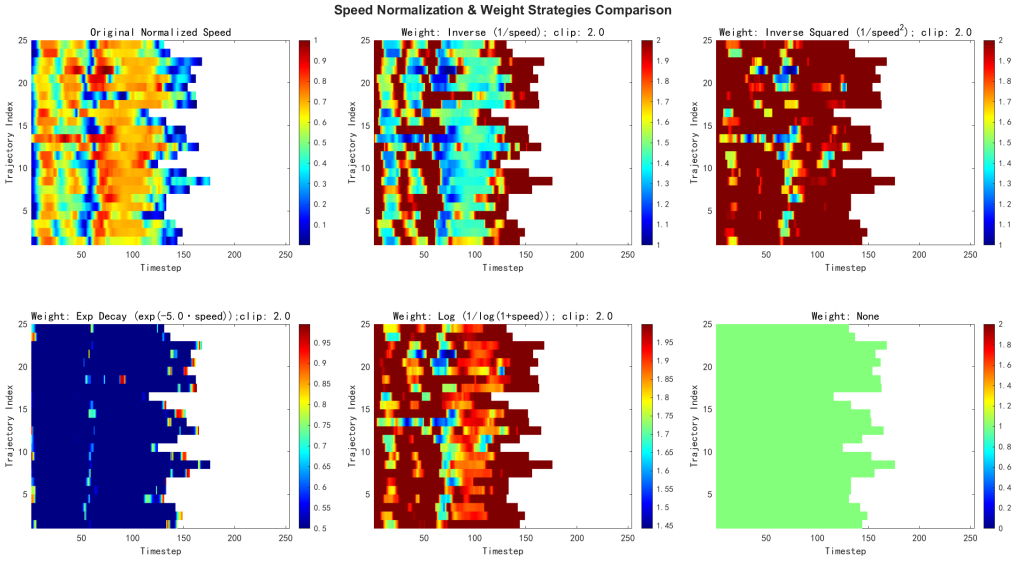

Figure 8: Visualization of Action Speed in LIBERO-OBJECT Task with Different $clip_{max} = 2.0$

actions and exert mild, localized suppression on fast actions. Within this group, the intensity of low-speed amplification follows a clear hierarchy: inverse squared (Equation 13) yields the strongest enhancement, followed by logarithmic weighting (Equation 15), and then inverse weighting (Equation 12). This consistent trend is observable across all clipping thresholds in Figures 7–10, validating the design principles of velocity-field-based action attention and supporting the selection of inverse squared weighting as the default configuration in the main experiments.

E Details about Model Training

This section presents comprehensive training and implementation details for AttenA+OFT (evaluated on LIBERO) and AttenA+WAM (evaluated on RoboTwin), covering architectural modifications,

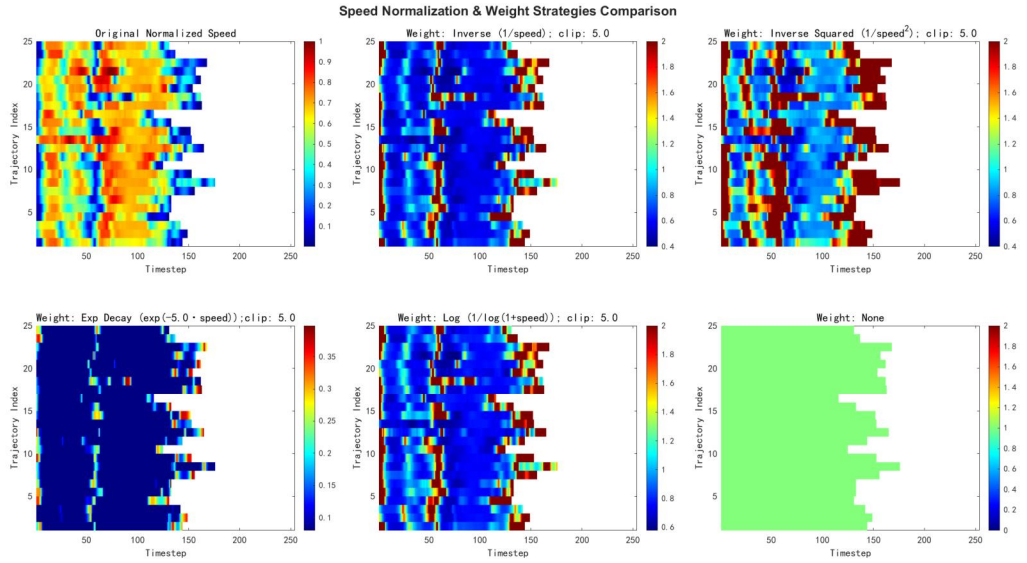

Figure 9: Visualization of Action Speed in LIBERO-OBJECT Task with Different $clip_{max} = 5.0$

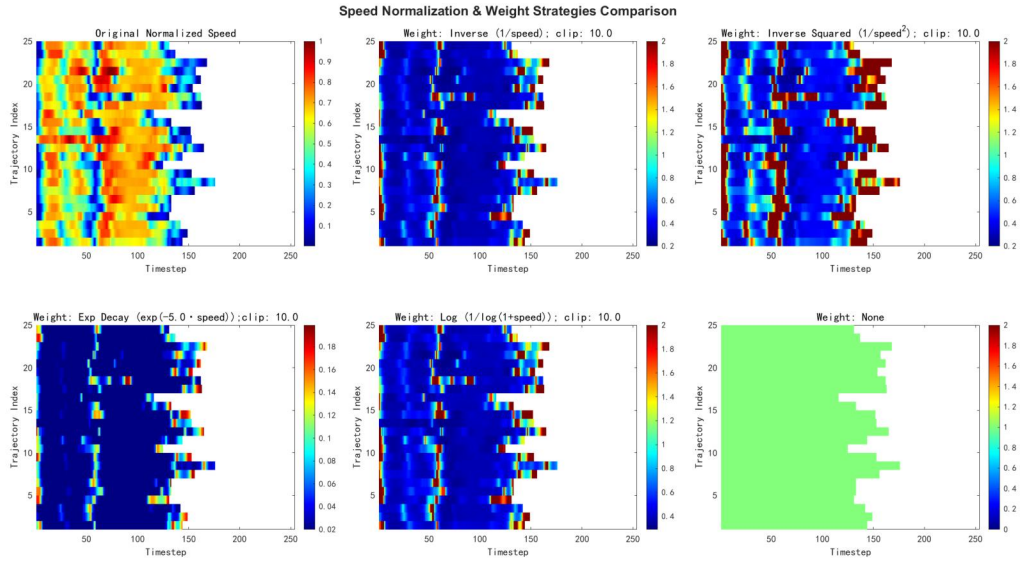

Figure 10: Visualization of Action Speed in LIBERO-OBJECT Task with Different $clip_{max} = 10.0$

optimization configurations, fine-tuning pipelines, checkpoint scheduling, and best-model selection criteria. All experiments in this work use the weight clipping settings $clip_{\max} = 2.0$ and $clip_{\max} = 5.0$, which are applied consistently across both model variants.

E.1 AttenA+OFT

We build AttenA+OFT as a direct adaptation of the OpenVLA-OFT framework, with our core velocity-field action attention integrated as a weighted module without altering the backbone architecture. Following the standard OpenVLA-OFT fine-tuning protocol, we train separate models for each of the four LIBERO task categories (Spatial, Object, Goal, Long) to ensure fair comparison with prior work. All models are trained for a total of $200{,}000$ steps, with checkpoints saved every $5{,}000$ steps. The taining time of each model using single H800 GPU is about 35 hours. During training, we strictly retain the original optimizer configuration, learning rate schedule, batch size, and data preprocessing used in OpenVLA-OFT to isolate the improvement brought by action attention. After training, we evaluate over saved checkpoints on the corresponding LIBERO test split and select the checkpoint with the highest success rate as the final model for reporting results.

E.2 AttenA+WAM

We implement AttenA+WAM on top of the Fast-WAM architecture, again integrating our velocity-guided action attention as a plug-and-play weighting module. Since the official Fast-WAM repository does not release end-to-end fine-tuning code, we adopt a practical and fair adaptation protocol: we freeze all vision encoders and the pre-trained WAM backbone, and only fine-tune the final action head using our proposed action attention mechanism. This design ensures we only introduce our method while preserving the pre-trained knowledge of the original model. We fine-tune on the RoboTwin dataset for 1 epoch, saving checkpoints every $2{,}000$ steps. The taining time of the model using two H800 GPUs is about 4 days. Consistent with AttenA+OFT, we evaluate over intermediate checkpoints and select the best-performing one based on validation success rate for final experimental comparisons.

F Detailed Evaluation Results on RoboTwin 2.0 and Real Franka Robot Experiments

F.1 Detailed Results on RoboTwin 2.0

Table 5: We present quantitative results on the RoboTwin 2.0 simulation benchmark, covering 50 bimanual manipulation tasks with two difficulty levels. RoboTwin 2.0 serves as a rigorous dual-arm manipulation testbed that demands precise bilateral coordination. The easy setting adopts fixed initial scene arrangements, whereas the hard setting introduces randomized object placements and scene configurations for higher generalization challenges.

Model	AttenA+WAM(Ours)		Fast-WAM		LingBot		Pi_05		Pi_0		X-VLA		Motus	
Task Type	clean	random	clean	random	clean	random	clean	random	clean	random	clean	random	clean	random
Adjust Bottle	100	100	100	100	90	94	100	99	99	95	100	99	89	93
Beat Block Hammer	99	93	99	97	96	98	96	93	79	84	92	88	95	88
Blocks Ranking RGB	100	100	100	100	99	98	92	85	80	63	83	83	99	97
Blocks Ranking Size	93	94	94	98	94	96	49	26	14	5	67	74	75	63
Click Alarmclock	100	100	100	100	99	100	98	89	77	68	99	99	100	100
Click Bell	100	100	100	100	100	100	99	66	71	48	100	100	100	100
Dump Bin Big Binbin	97	95	97	96	89	96	92	97	88	83	79	77	95	91
Grab Roller	100	100	100	100	100	100	100	100	98	94	100	100	100	100
Handover Block	95	90	95	81	99	78	66	57	47	31	73	37	86	73
Handover Mic	100	91	99	100	94	96	98	97	97	97	0	0	78	63
Hanging Mug	67	62	58	62	40	28	18	17	14	11	23	27	38	38
Lift Pot	100	100	100	100	100	99	96	85	80	72	99	100	96	99
Move Can Pot	89	91	90	88	94	97	51	55	68	48	89	86	34	74
Move Pillowbottle Pad	98	100	100	99	99	99	84	61	67	46	73	71	93	96
Move Playingcard Away	100	100	100	100	100	99	96	84	74	65	93	98	100	96
Move Stapler Pad	74	70	77	64	91	79	56	42	41	24	78	73	83	85
Open Laptop	99	100	98	100	92	94	90	96	71	81	93	100	95	91
Open Microwave	71	49	62	45	82	86	34	77	4	32	79	71	95	91
Pick Diverse Bottles	91	85	80	85	89	82	81	71	69	31	58	36	90	91
Pick Dual Bottles	100	96	100	96	100	99	93	63	59	37	47	36	96	90
Place A2B Left	97	96	95	93	97	93	87	82	43	47	48	49	82	79
Place A2B Right	95	98	93	99	97	95	87	84	39	34	36	36	90	87
Place Bread Basket	93	93	91	93	97	95	77	64	62	46	81	71	91	94
Place Bread Skillet	88	91	90	93	95	90	85	66	66	49	77	67	86	83
Place Burger Fries	96	96	96	99	97	95	94	87	81	76	94	94	98	98
Place Can Basket	65	64	71	69	81	84	62	62	55	46	49	52	81	76
Place Cans Plasticbox	99	100	99	96	100	99	94	84	63	45	97	98	98	94
Place Container Plate	98	99	96	100	99	97	99	95	97	92	97	95	98	99
Place Dual Shoes	86	89	94	88	94	89	75	75	59	51	79	88	93	87
Place Empty Cup	99	100	100	100	100	100	100	99	91	85	100	98	99	98
Place Fan	97	91	96	96	99	93	87	85	66	71	80	75	91	87
Place Mouse Pad	88	88	83	89	93	96	60	39	20	20	70	70	66	68
Place Object Basket	91	85	89	88	91	88	80	76	67	70	44	39	81	87
Place Object Scale	92	94	90	97	96	95	86	80	57	52	52	74	88	85
Place Object Stand	90	91	90	94	99	96	91	85	82	68	86	88	98	97
Place Phone Stand	99	99	97	99	97	97	81	81	49	53	88	87	87	86
Place Shoe	96	99	96	99	98	98	92	93	76	76	96	95	99	97
Press Stapler	92	94	90	97	85	82	87	83	44	37	92	98	93	98
Put Bottles Dustbin	91	91	95	90	87	91	84	79	65	56	74	77	81	79
Put Object Cabinet	85	85	94	89	85	87	80	79	73	60	46	48	88	71
Rotate QRcode	92	91	93	89	96	91	89	87	74	70	34	33	89	73
Scan Object	93	89	89	92	96	91	72	65	55	42	14	36	67	66
Shake Bottle Horizontally	100	100	100	100	100	99	99	99	98	92	100	100	100	98
Shake Bottle	100	100	100	100	100	97	99	97	94	91	99	100	100	97
Stack Blocks Three	97	96	95	97	99	98	91	76	72	52	6	10	91	95
Stack Blocks Two	100	100	100	100	100	98	97	100	93	79	92	87	100	98
Stack Bowls Three	95	95	80	81	86	83	77	71	77	75	76	86	79	87
Stack Bowls Two	95	95	92	98	94	98	95	96	94	95	96	93	98	98
Stamp Seal	91	89	90	94	96	97	79	55	46	33	76	82	93	92
Turn Switch	80	79	61	59	44	45	62	54	41	42	40	61	84	78
Average	93.06	91.86	91.88	91.78	92.9	91.5	82.74	76.76	65.92	58.4	72.88	72.84	88.52	87.02

Table 6: Detailed Evaluation Results on Real Franka Robot Experiments. We report the success rates (SR) across four manipulation tasks under the Baseline and our Attention-based method.

Experiment	Baseline				AttenA+			
	Close Draw	Put Cube	Multi-object	Long	Close Draw	Put Cube	Multi-object	Long
1	1	1	1	1	1	1	1	1
2	1	1	1	0	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1
6	1	1	0	1	1	1	1	1
7	1	1	1	0	1	1	1	1
8	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	0
10	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1
12	1	0	1	0	1	1	1	1
13	1	1	1	0	1	1	1	1
14	1	1	0	1	1	1	1	1
15	1	1	1	1	1	1	1	1
16	1	1	0	1	1	1	1	1
17	1	1	1	1	1	1	1	1
18	1	1	1	0	1	1	1	1
19	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	0
23	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	0	1
30	1	1	1	1	1	1	1	0
31	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	1
33	1	1	1	1	1	1	1	1
34	1	1	0	1	1	1	1	1
35	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1
37	1	1	1	1	1	1	1	1
38	1	0	1	1	1	1	1	1
39	1	1	1	0	1	1	1	1
40	1	1	1	1	1	1	1	1
41	1	1	1	1	1	1	1	0
42	1	1	1	1	1	1	1	1
43	1	1	0	0	1	1	1	1
44	1	1	1	1	1	1	1	0
45	1	1	1	1	1	1	1	1
46	1	1	1	1	1	1	1	1
47	1	1	1	0	1	1	1	1
48	1	1	1	1	1	1	1	1
49	1	1	1	1	1	1	1	1
50	1	1	1	1	1	1	1	1
SR (%)	**100**	**96**	**90**	**84**	**100**	**100**	**98**	**90**

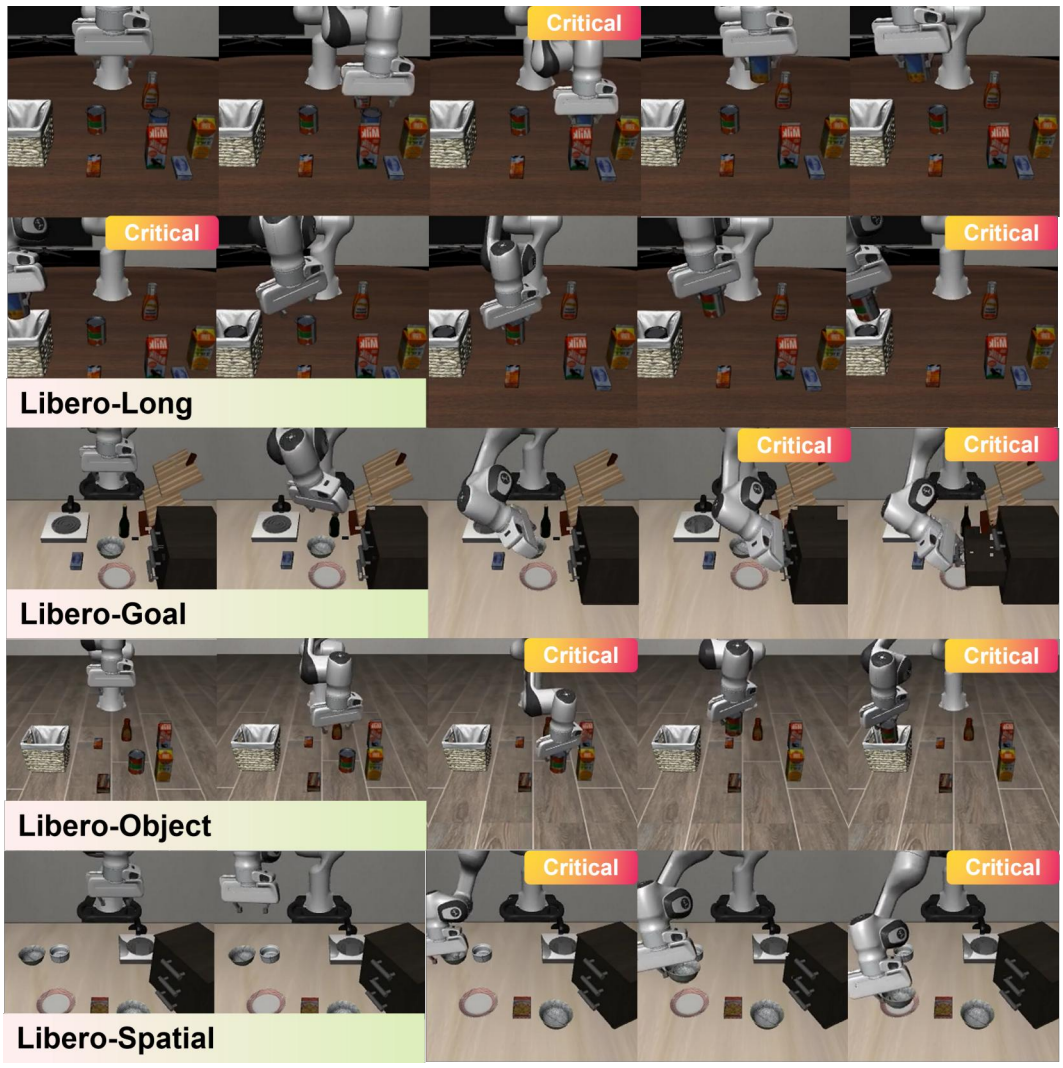

Figure 11: Third-person views of example LIBERO manipulation tasks. Frames labeled 'critical' highlight slow, high-precision actions (e.g., grasping, alignment) where AttenA+ applies increased attention weights to improve task success.

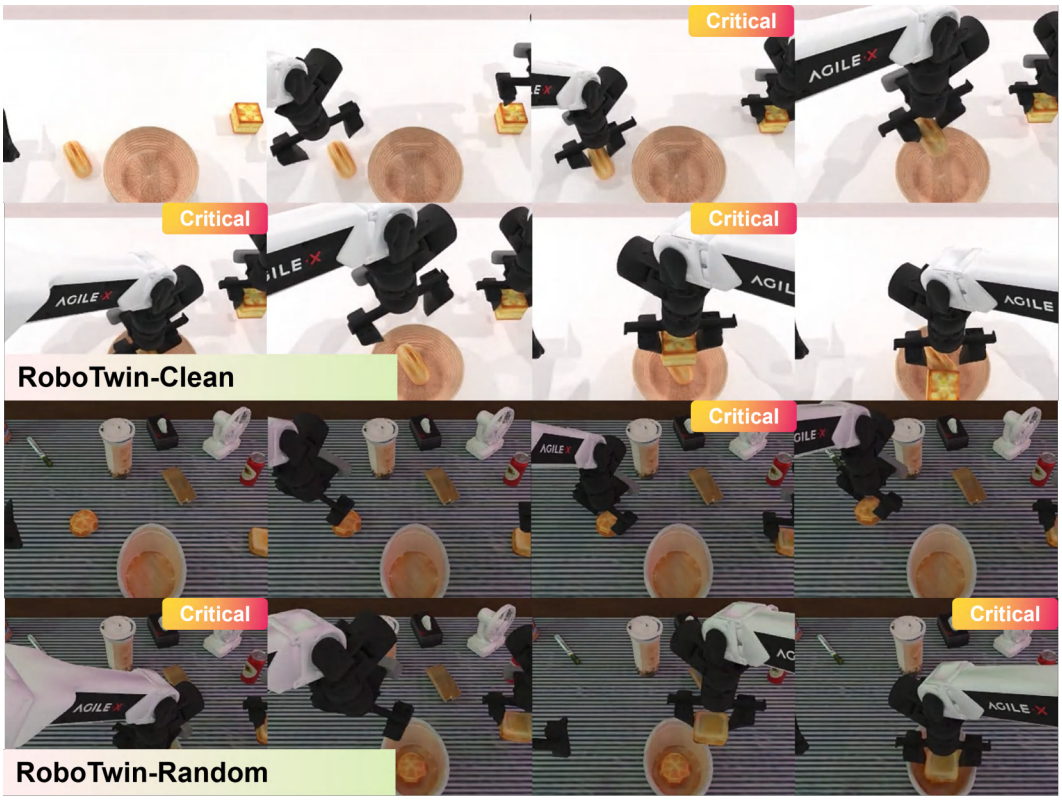

Figure 12: Third-person views of example RoboTwin tasks in both clean and randomized environments. The 'critical' labels mark slow, precision-sensitive steps, where AttenA+ prioritizes learning to boost performance across diverse conditions.

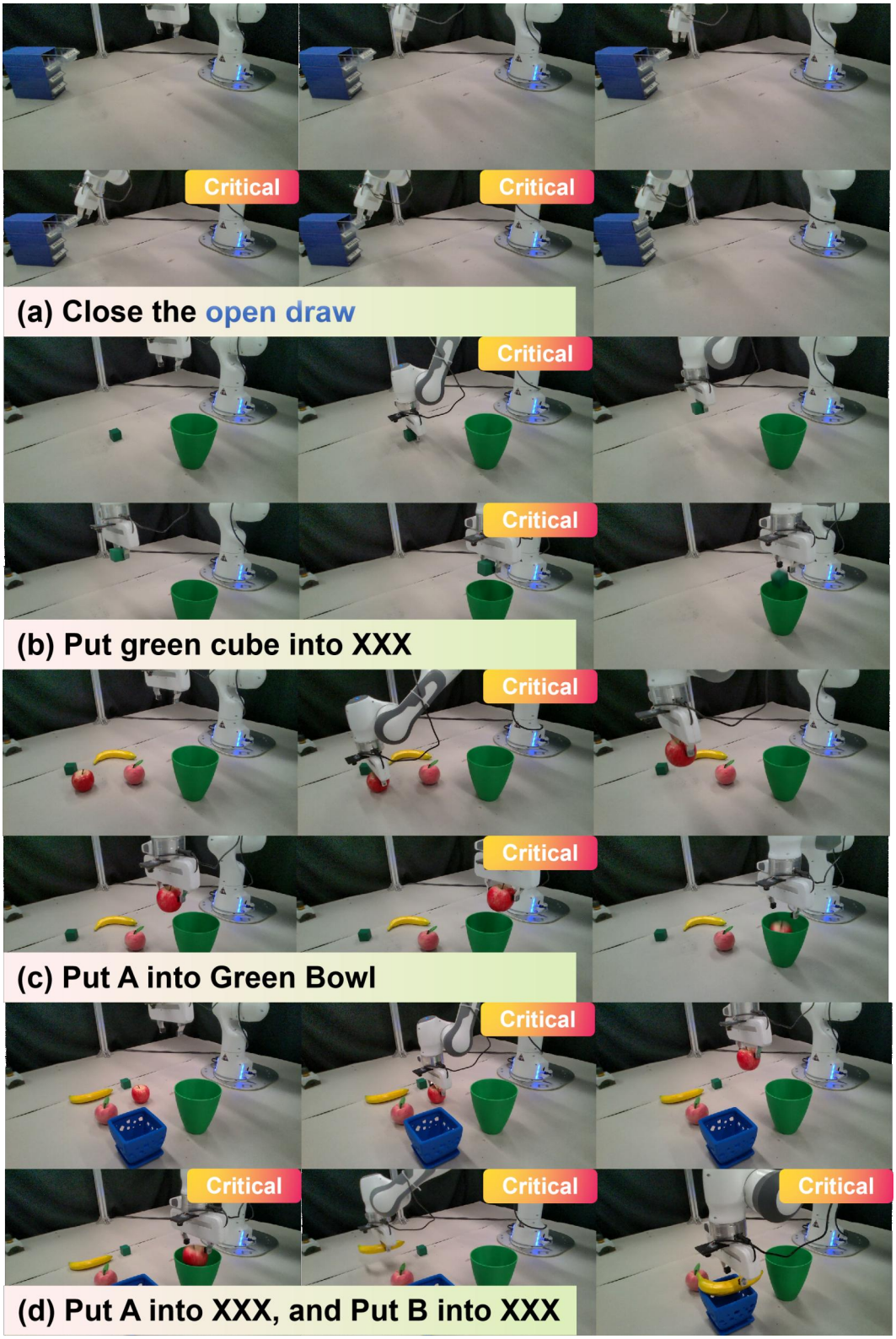

Figure 13: Third-person views of four representative real-world Franka tasks. The 'critical' labels identify slow, high-precision manipulation steps, demonstrating how AttenA+ prioritizes these phases to improve real-robot success rates.