ATI-VLA: Action-Centric Predictive Vision–Language–Action Models via Actionable Alignment Then Adaptive Injection

Yijie Zhu[1,2], **Rui Shao**[1,5]*, **Jie He**[1], **Wei Li**[1], **Bo Zhao**[2], **Yelin Wang**[2], **Xiaochen Yuan**[3], **Tao Tan**[3], **Miao Zhang**[1], **Xiaojiang Peng**[4], **Zitong Yu**[2,6]*

[1]Harbin Institute of Technology, Shenzhen
[2]Institute for Artificial Intelligence, Great Bay University [3]Macao Polytechnic University
[4]Shenzhen Technology University [5]Shenzhen Loop Area Institute
[6]Dongguan Key Laboratory for Intelligence and Information Technology
zyj99_hitsz@stu.hit.edu.cn shaorui@hit.edu.cn yuzitong@gbu.edu.cn
https://jiutian-vl.github.io/ATI-VLA-page

Abstract

Predictive Vision–Language–Action (VLA) models aim to improve robotic manipulation via future observation or world dynamics forecasting. However, existing approaches often fail to realize this potential and underperform direct action prediction models. We argue that these limitations stem from modality misalignment between observations and actions, together with joint optimization conflicts that drive learning away from an action-centric objective. To this end, we introduce ***ATI-VLA***, an Action-Centric Predictive Vision–Language–Action framework via Actionable ***Alignment*** Then Adaptive ***Injection***. Specifically, it follows a two-step design: **1)** ***Actionable Representation Alignment via a Shared Codebook*.** It aligns predictive observation and action representations by mapping both modalities into a shared discrete latent space via a unified codebook, making predictive observation latents readily usable for action generation and mitigating modality misalignment. **2)** ***Action-Centric Adaptive Injection of Predictive Latents***. Building upon this, it then injects predictive observation latents into action decoding as explicit predictive priors via a ***lightweight adaptive side-path***, enabling adaptive predictive guidance under a single action-centric objective. Extensive experiments on both simulation and real-world robotic tasks demonstrate that ATI-VLA achieves state-of-the-art performance with faster convergence.

1 Introduction

Vision–Language–Action (VLA) models have emerged as a unified paradigm for robotic manipulation, traditionally mapping observations and language instructions directly to low-level actions [18, 59, 27, 4, 32, 51, 52, 29], as shown in Fig. 1(a). More recently, predictive VLA [25, 73, 69, 67] approaches extend this paradigm by explicitly forecasting future observations or world dynamics to provide anticipatory guidance for action generation, as shown in Fig. 1(b).

However, our analysis reveals that existing predictive VLA approaches generally underperform non-predictive VLA baselines, as illustrated in Fig. 1(d). This indicates that introducing future prediction as an auxiliary task does not reliably translate into effective improvements in action generation. We argue that this performance gap arises because these predictive approaches fail to maintain an

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

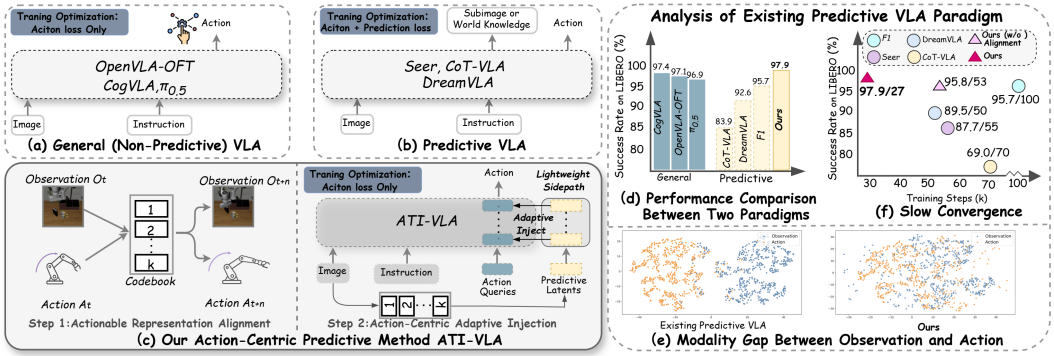


Figure 1: Overview of VLA paradigms and motivation for ***ATI-VLA***. ***Figures (a)–(c)*** compare general (non-predictive) VLA, predictive VLA, and our proposed action-centric predictive framework, ***ATI-VLA***. ***Figures (d)–(f)*** identify key limitations of existing predictive VLA approaches, including observation–action modality gaps and slow convergence induced by joint optimization, which are responsible for their inferior performance relative to general VLA. By addressing these limitations, ATI-VLA enables the predictive VLA paradigm to more fully realize its potential, achieving state-of-the-art performance with faster convergence.

action-centric learning focus, primarily due to two factors: ***i) The Modality Gap Between Visual Observation and Action.*** As shown in Fig. 1(e), latent distributions analysis of recent predictive VLA models [67, 45] reveals a pronounced modal gap between observation prediction and action generation. This gap weakens the grounding of predictive representations in action generation, thereby diluting the action-centric learning focus. ***ii) Optimization Conflicts Induced by Joint Training.*** Existing predictive VLA paradigms typically introduce future prediction as an auxiliary task by directly adding its loss to the action objective. As illustrated in Fig. 1(f), such joint optimization leads to unstable training dynamics and slow convergence. Moreover, competing gradients from prediction and action objectives can divert optimization away from action-centric objectives and, in extreme cases, result in training collapse [67]. Together, these factors hinder predictive VLA approaches from fully realizing their performance potential. **We provide a more in-depth analysis of these two issues in Appendix C.2.**

To address these challenges, we propose ***ATI-VLA***, an action-centric predictive VLA framework via Actionable ***Alignment*** Then Adaptive ***Injection***, explicitly aligning predictive representations with action generation and injecting predictive information under a single action-centric objective. Specifically, as shown in Fig. 1(c), ATI-VLA follows a two-step design: ***1) Actionable Representation Alignment via a Shared Codebook.*** To bridge the modal gap between predictive observations and action generation, ATI-VLA introduces an actionable representation alignment via a shared codebook. By mapping both modalities into a unified discrete latent space, predictive representations are directly grounded in the action space, making them readily compatible with the action decoding process rather than being implicitly entangled through joint training as in prior approaches. ***2) Action-Centric Adaptive Injection of Predictive Latents.*** Building upon the aligned and actionable representations, ATI-VLA further introduces an action-centric adaptive injection mechanism to integrate predictive information into action generation. To enable predictive guidance to adapt as action generation progresses, we employ a lightweight adaptive side-path that modulates the action decoding process conditioned on predictive latents. Moreover, to avoid predictive information from overly interfering with action decoding, this modulation is applied at regular layer intervals, allowing predictive guidance while preserving fine-grained control. Crucially, this injection is optimized solely with the action loss, ensuring that predictive information serves purely as guidance without inducing optimization conflicts.

Extensive experimental results demonstrate that, by addressing the aforementioned challenges, ATI-VLA is able to fully unlock the potential of predictive VLA paradigms. Across both simulation benchmarks and real-world robotic tasks, ATI-VLA consistently achieves state-of-the-art performance, while exhibiting faster convergence compared to existing predictive approaches. To summarize, our main contributions are as follows:

- We propose ***ATI-VLA***, an action-centric predictive VLA framework with an ***Align-Then-Inject*** design, which effectively addresses the lack of action-centric learning in existing predictive paradigms.
- We introduce an ***actionable representation alignment*** via a shared codebook to ground predictive observations directly in the action space.
- We design an ***action-centric adaptive injection*** strategy that integrates predictive information into action decoding via a lightweight side-path under a single action-centric objective.
- Extensive experiments validate the effectiveness of ATI-VLA, achieving state-of-the-art performance with faster convergence.

2 Method

2.1 Preliminary: Predictive VLA

The Overall Pipeline. Predictive VLA [67, 69, 30, 45] models improve robot execution by using future perceptual or world knowledge prediction to guide action generation. At each time step t, the robot receives a set of heterogeneous inputs, including a natural language instruction $\mathcal{I}$, a visual observation $\mathbf{o}_t$, and two types of special query tokens: the $<img>$ token for future prediction and the $<act>$ token for action generation. All inputs are concatenated into a unified token sequence and processed by a multimodal model $\mathcal{H}$ to produce a predicted latent that encodes future visual content or world knowledge:

$$\mathbf{z}_{t+n}^{\text{pred}} = \mathcal{H}\big(\mathbf{o}_t, \mathcal{I} \mid <img>\big). \tag{1}$$

Then, a decoder $\mathcal{D}_{\text{img}}$ maps this latent back to the target prediction space, producing either the future image $\hat{\mathbf{o}}_{t+n}$ or the future world knowledge $\hat{\mathbf{w}}_{t+n}$ at time step $t+n$:

$$\hat{\mathbf{y}}_{t+n} = \mathcal{D}_{\text{img}}\Big(\mathbf{z}_{t+n}^{\text{pred}}\Big), \hat{\mathbf{y}}_{t+n} \in \{\hat{\mathbf{o}}_{t+n},\ \hat{\mathbf{w}}_{t+n}\}. \tag{2}$$

For action generation, given the predicted latent $\mathbf{z}_{t+n}^{\text{pred}}$, the $<act>$ query is mapped by the unified model $\mathcal{H}$ to a latent action embedding that aggregates action-relevant information from the unified token sequence. Finally, an action head $\mathcal{D}_{\text{act}}$ maps the latent feature to an n-step action chunk:

$$\mathbf{A}_{t:t+n-1} = \mathcal{D}_{\text{act}}\Big(\mathcal{H}\big(\mathbf{o}_t, \mathcal{I}, <img>| <act>\big)\Big). \tag{3}$$

During training, predictive VLA models are typically optimized with a joint objective that supervises both action generation and future prediction:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda\, \mathcal{L}_{\text{pred}}, \tag{4}$$

where λ balances action and prediction supervision.

Problem Definition. As discussed above, this design often introduces a modality gap between predictive and action representations, along with competing optimization objectives that weaken action-centric learning. As a result, predictive information may fail to effectively support action generation and can degrade training stability. The central problem we address is how to leverage predictive representations to enhance action generation while preserving action-centric optimization.

2.2 ATI-VLA

2.2.1 OVERVIEW

In this section, we present ATI-VLA, an ***Alignment Then Injection*** framework that couples Actionable Representation Alignment with Action-Centric Adaptive Injection. ATI-VLA is trained in two stages (Algorithm 1): we first optimize **Actionable Representation Alignment** to learn a unified, actionable latent space via a shared codebook, making predictive observation latents directly usable for action generation and mitigating modality misalignment. We then freeze the Alignment module and train **Action-Centric Adaptive Injection** with a lightweight adaptive side-path to inject predictive latents into action decoding, optimizing only the action objective. We next elaborate on the two steps in turn, including their architectures and training objectives.

Algorithm 1 ATI-VLA: Align-Then-Inject Framework

Require: Current observation and action $\mathcal{O}_t, \mathcal{A}_t$; future observation and action $\mathcal{O}_{t+n}, \mathcal{A}_{t+n}$ (n: chunk size); instruction $\mathcal{I}$, initial action tokens $\mathbf{T}_{\text{act}}$; injection frequency s, action length l_a; Large language model $\mathbf{H}$ (L layers, layer-l hidden states $\mathbf{H}_l$); layer-l injection weight β_l.

Step 1: Actionable Representation Alignment

$\mathbf{Z}_{\mathcal{O}_t} \leftarrow \text{Quant}(\text{Encode}(\mathcal{O}_t)),\ \mathbf{Z}_{\mathcal{A}_t} \leftarrow \text{Quant}(\text{Encode}(\mathcal{A}_t))$ *// Quantize both modalities via a shared codebook*

$\hat{\mathcal{O}}_{t+n}, \hat{\mathcal{A}}_{t+n} \leftarrow \text{UnifiedDecoder}([\mathcal{O}_t, \mathcal{A}_t], [\mathbf{Z}_{\mathcal{O}_t}, \mathbf{Z}_{\mathcal{A}_t}])$

$\mathcal{L}_{\text{align}} \leftarrow \lambda_{\mathcal{O}} \mathcal{L}_{\text{obs}}(\mathcal{O}_{t+n}, \hat{\mathcal{O}}_{t+n}) + \lambda_{\mathcal{A}} \mathcal{L}_{\text{act}}(\mathcal{A}_{t+n}, \hat{\mathcal{A}}_{t+n})$; update θ_{Align}; freeze trained θ_{Align} after Step 1

Step 2: Action-Centric Adaptive Injection

$\mathbf{Z}_{\mathcal{O}_t} \leftarrow \text{Align}(\mathcal{O}_t),\ \mathbf{H}_0 \leftarrow [\mathcal{O}_t, \mathcal{I}, \mathbf{T}_{\text{act}}]$

for $l = 1$ **to** L **do**

  $\mathbf{H}_l \leftarrow \text{LLMLayer}_l(\mathbf{H}_{l-1})$

  if $l \bmod s = 0$ **then**

    $\mathbf{H}_l^{\text{act}} \leftarrow \mathbf{H}_l[:, -l_a:, :],\ \mathbf{Z}_{\mathcal{O}_t} \leftarrow \mathbf{Z}_{\mathcal{O}_t} + \text{FFN}(\mathbf{Z}_{\mathcal{O}_t})$ *// $\mathbf{Z}_{\mathcal{O}_t}$ is transformed via side-path for injection*

    $\mathbf{H}_l^{\text{cross}} \leftarrow \text{CrossAttn}(\mathbf{H}_l^{\text{act}}, \mathbf{Z}_{\mathcal{O}_t}),\ \mathbf{H}_l[:, -l_a:, :] \leftarrow \mathbf{H}_l^{\text{act}} + \beta_l \mathbf{H}_l^{\text{cross}}$

  end if

end for

$\hat{\mathcal{A}}_{t:t+n-1} \leftarrow \text{ActionHead}(\mathbf{H}_L[:, -l_a:, :]),\ \mathcal{L}_{\text{action}} \leftarrow \mathcal{L}_{\text{act}}(\mathcal{A}_{t:t+n-1}, \hat{\mathcal{A}}_{t:t+n-1})$

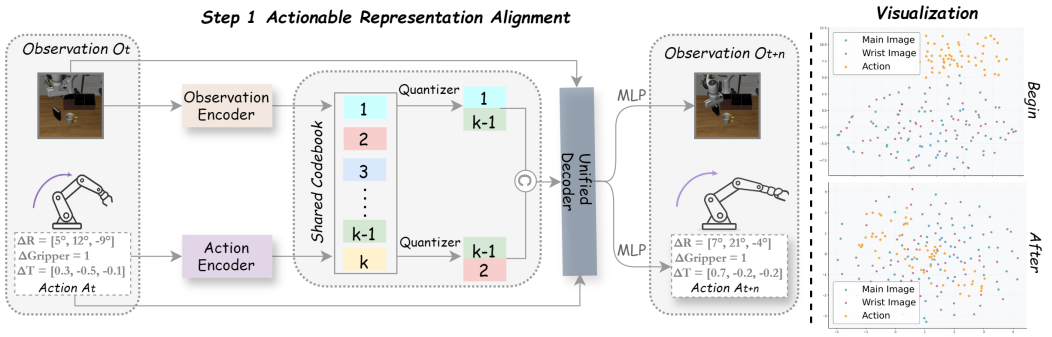


Figure 2: **Step 1: Actionable Representation Alignment. Left:** The current observation and action are encoded and vector-quantized into a shared codebook, then decoded by a unified attention-based decoder to jointly predict the future observation $\mathcal{O}_{t+n}$ and action $\mathcal{A}_{t+n}$. **Right:** t-SNE visualization of observation (main/wrist) and action latents during training: they are initially dispersed and modality-separated, but progressively converge into an aligned shared structure after training.

2.2.2 Actionable Representation Alignment

As discussed above, we first bridge the observation–action modality gap in predictive VLA via an explicit alignment step. Recent works [10, 5] have explored extracting action semantics from consecutive frames into compact latent representations, motivating structured cross-modal abstractions. Inspired by these efforts, we propose Actionable Representation Alignment via a Shared Codebook to bridge this modality gap, as illustrated in Fig. 2. To ensure effective alignment, we enforce temporal consistency and causal alignment between observation and action representations by adopting a symmetric, predictive design across both branches. Predictive and action representations are mapped into a shared discrete latent via a unified codebook, grounding predictive latents in the action space.

Modality-Specific Encoding. Specifically, given the current observation–action pair $(\mathcal{O}_t, \mathcal{A}_t)$ and the future pair $(\mathcal{O}_{t+n}, \mathcal{A}_{t+n})$, where n denotes the action chunk size, we encode $\mathcal{O}_t$ and $\mathcal{A}_t$ using modality-specific encoders $f_{\mathcal{O}}(\cdot)$ and $f_{\mathcal{A}}(\cdot)$. To explicitly align the two modalities, we introduce a shared codebook $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^{K} \subset \mathbb{R}^d$, which defines a unified discrete latent space. Given a continuous embedding $\mathbf{z} \in \mathbb{R}^d$, the vector quantization operator $\text{Quant}(\cdot)$ maps it to the nearest codeword in $\mathcal{C}$:

$$\text{Quant}(\mathbf{z}) = \mathbf{c}_{k^*},\ k^* = \arg\min \|\mathbf{z} - \mathbf{c}_k\|_2^2. \tag{5}$$

Using this shared quantization process, the encoded observation and action representations are mapped into the same discrete latent space:

$$\mathbf{Z}_{\mathcal{O}_t} = \text{Quant}(f_{\mathcal{O}}(\mathcal{O}_t)),\quad \mathbf{Z}_{\mathcal{A}_t} = \text{Quant}(f_{\mathcal{A}}(\mathcal{A}_t)). \tag{6}$$

Unified Decoding. Given the discrete latent representations $\mathbf{Z}_{\mathcal{O}_t}$ and $\mathbf{Z}_{\mathcal{A}_t}$ obtained from the shared codebook, we employ a unified decoder composed of stacked self-attention and cross-attention layers to jointly model future observations and actions. We first construct the input token sequence by concatenating the current observation and action, denoted as $\mathbf{X}_t = [\mathcal{O}_t, \mathcal{A}_t]$. These tokens are fed into self-attention to capture intra- and cross-modal dependencies:

$$\mathbf{H}_t^{\text{self}} = \text{SelfAttn}(\mathbf{X}_t) = \text{Softmax}\left(\frac{\mathbf{Q}_t \mathbf{K}_t^\top}{\sqrt{d}}\right)\mathbf{V}_t, \tag{7}$$

where $\mathbf{Q}_t$, $\mathbf{K}_t$, and $\mathbf{V}_t$ are the query, key, and value projections of $\mathbf{X}_t$, respectively. Next, the self-attended representation $\mathbf{H}_t^{\text{self}}$ is used as the query in a cross-attention block, which attends to the concatenated discrete latents $\mathbf{Z}_t = [\mathbf{Z}_{\mathcal{O}_t}, \mathbf{Z}_{\mathcal{A}_t}]$ serving as keys and values:

$$\mathbf{H}_{t+n} = \text{CrossAttn}(\mathbf{H}_t^{\text{self}}, \mathbf{Z}_t, \mathbf{Z}_t). \tag{8}$$

The resulting hidden representation $\mathbf{H}_{t+n}$ encodes predictive, action-aware semantics grounded in the shared discrete latent space. Two lightweight projection heads, $\text{MLP}_{\mathcal{O}}$ and $\text{MLP}_{\mathcal{A}}$, are then applied to map $\mathbf{H}_{t+n}$ to the future observation and action predictions, respectively:

$$\hat{\mathcal{O}}_{t+n} = \text{MLP}_{\mathcal{O}}(\mathbf{H}_{t+n}),\ \hat{\mathcal{A}}_{t+n} = \text{MLP}_{\mathcal{A}}(\mathbf{H}_{t+n}). \tag{9}$$

Training Objective. Specifically, the training objective consists of two components: an observation prediction loss and an action prediction loss. The overall alignment objective is defined as

$$\mathcal{L}_{\text{align}} = \lambda_{\mathcal{O}}\mathcal{L}_{\text{obs}}\big(\hat{\mathcal{O}}_{t+n}, \mathcal{O}_{t+n}\big) + \lambda_{\mathcal{A}}\mathcal{L}_{\text{act}}\big(\hat{\mathcal{A}}_{t+n}, \mathcal{A}_{t+n}\big), \tag{10}$$

where $\mathcal{L}_{\text{obs}}$ and $\mathcal{L}_{\text{act}}$ measure errors for observation and action prediction, and $\lambda_{\mathcal{O}}, \lambda_{\mathcal{A}}$ balance the two terms. This objective encourages the shared discrete latent space to encode temporally consistent, causally aligned semantics. As shown in Fig. 2, the observation and action latents are initially dispersed and modality-separated, but progressively co-locate and consolidate into a coherent shared structure during Alignment training. This trend suggests that our method effectively reduces cross-modal misalignment and grounds predictive latents in action-relevant semantics. **Beyond empirical evidence, we also provide theoretical grounding for this design in Appendix D.1.**

2.2.3 Action-Centric Adaptive Injection

Design Rationale. With actionable and predictive representations learned in Alignment, we next integrate predictive information into action generation. We use predictive latents purely as guidance while retaining a single action-centric objective. A straightforward choice is to inject predictive signals at every decoding layer. **However, our theoretical analysis (Appendix D.2) and empirical results (Tab. 6)** show that dense, layer-wise injection can over-constrain decoding, degrading control precision and destabilizing training. This reflects that predictive cues are not uniformly beneficial across the decoding hierarchy: early layers primarily benefit from coarse semantic guidance, whereas later layers require fine-grained motor control. We therefore introduce predictive information *adaptively*, rather than coupling it to every decoding step.

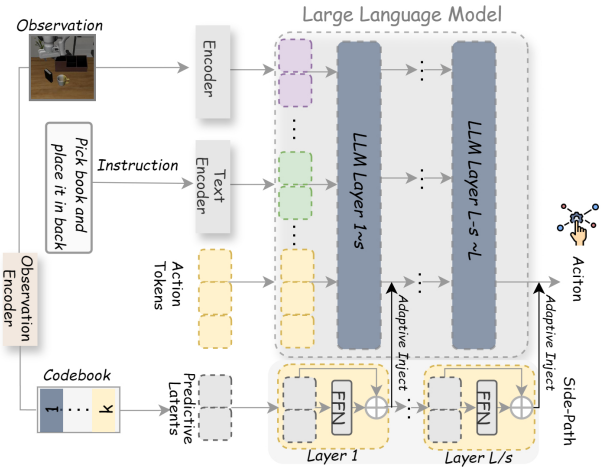


Figure 3: **Step 2: Action-Centric Adaptive Injection.** Predictive observation latents are adaptively transformed via a lightweight side-path and injected into LLM layers to modulate action query tokens. Interval-based injection provides effective predictive guidance while preserving stable, action-centric decoding.

Adaptive Side-Path Injection. As illustrated in Fig. 3, we introduce a lightweight adaptive side-path that operates in parallel with a large language model (LLM) backbone, enabling predictive information to guide action generation without interfering with the core action-centric optimization. At each time step t, the LLM

Table 1: **Simulation Results on LIBERO.** Comparison of task success rates and their ranks.

Category	Method	Success Rate(↑)					Rank(↓)
		Spatial	Object	Goal	Long	Average	
General VLA	OpenVLA-OFT *[RSS'25]* [20]	97.6%	98.4%	**97.9%**	94.5%	97.1%	2
	π_0 *[RSS'25]* [2]	96.8%	98.8%	95.8%	85.2%	94.2%	7
	PD-VLA *[IROS'25]*† [44]	95.5%	96.7%	94.9%	91.7%	94.7%	6
Predictive VLA	ATM *[RSS'24]* [50]	68.5%	68.0%	77.8%	39.3%	63.4%	11
	Seer *[ICLR'25]* [45]	-	-	-	87.7%	-	-
	DreamVLA *[NeurIPS'25]* [67]	97.5%	94.0%	89.5%	89.5%	92.6%	8
	CoT-VLA *[CVPR'25]* [69]	87.5%	91.6%	87.6%	69.0%	83.9%	9
	$\mathcal{F}_1$ *[arXiv'25]* [35]	98.2%	97.8%	95.4%	91.3%	95.7%	3
	H-GAR *[AAAI'26]* [73]	-	-	-	94.0%	-	-
	UniVLA *[RSS'25]* [5]	96.5%	96.8%	95.6%	92.0%	95.2%	5
	WorldVLA *[arXiv'25]* [7]	87.6%	96.2%	83.4%	60.0%	81.8%	10
	UniVLA *[ICLR'26]* [49]	95.4%	98.8%	93.6%	94.0%	95.5%	4
Predictive VLA	ATI-VLA (Ours)	**98.6%**	**99.4%**	97.2%	**96.2%**	**97.9%**	**1**

receives the following inputs: the current visual observation $\mathcal{O}_t$, a natural language instruction $\mathcal{I}$, and a set of learnable action query tokens $\mathbf{T}_{\text{act}}$ with length l_a, which are responsible for generating an n-step action chunk. These inputs are first embedded and concatenated to form the initial sequence:

$$\mathbf{H}_0 = \Big[f_{\mathcal{O}}(\mathcal{O}_t),\, f_{\mathcal{I}}(\mathcal{I}),\, \mathbf{T}_{\text{act}}\Big], \tag{11}$$

where $f_{\mathcal{O}}(\cdot)$ and $f_{\mathcal{I}}(\cdot)$ denote the observation and instruction encoders, respectively. The sequence is then fed into L stacked LLM Layers, yielding hidden states iteratively:

$$\mathbf{H}_l = \text{LLMLayer}_l(\mathbf{H}_{l-1}), \qquad l = 1, \ldots, L. \tag{12}$$

In parallel, we obtain the predictive latent $\mathbf{Z}_{\mathcal{O}_t}$ by encoding $\mathcal{O}_t$ and quantizing it with the shared codebook. This latent encodes predictive observation information and serves as the sole source of look-ahead guidance for the LLM. Before each injection, the predictive latent is updated as:

$$\mathbf{Z}_{\mathcal{O}_t}^{(l)} = \mathbf{Z}_{\mathcal{O}_t}^{(l-1)} + \text{FFN}\Big(\mathbf{Z}_{\mathcal{O}_t}^{(l-1)}\Big), \tag{13}$$

with $\mathbf{Z}_{\mathcal{O}_t}^{(0)} = \mathbf{Z}_{\mathcal{O}_t}$. This transformation adapts the predictive representation to the abstraction level of different LLM layers, instead of keeping it fixed.

Rather than injecting predictive information at every LLM layer, we perform injection at regular layer intervals to avoid over-constraining the decoding process. Concretely, given an injection frequency s, predictive guidance is applied only when $l \bmod s = 0$. At an injection layer l, we extract the hidden states corresponding to the action query tokens from the LLM output, denoted as $\mathbf{H}_l^{\text{act}} \in \mathbb{R}^{l_a \times d}$. The transformed predictive latent $\mathbf{Z}_{\mathcal{O}_t}^{(l)}$ is then injected via cross-attention:

$$\mathbf{H}_l^{\text{cross}} = \text{CrossAttn}\big(\mathbf{H}_l^{\text{act}}, \mathbf{Z}_{\mathcal{O}_t}^{(l)}, \mathbf{Z}_{\mathcal{O}_t}^{(l)}\big), \tag{14}$$

where the action token representations serve as queries, and the predictive latent provides keys and values. The action-related hidden states are subsequently modulated through a residual update:

$$\mathbf{H}_l^{\text{act}} \leftarrow \mathbf{H}_l^{\text{act}} + \beta_l \cdot \mathbf{H}_l^{\text{cross}}, \tag{15}$$

with β_l being a learnable scalar that controls the strength of predictive guidance at layer l. After the final LLM layer, the action query representations $\mathbf{H}_L^{\text{act}}$ are mapped to an n-step action chunk. Crucially, this stage is trained with only the action loss, without any auxiliary prediction objective. **Moreover, the subsequent empirical results show that the injection frequency s exhibits cross-backbone generalization and introduces little tuning burden.**

3 Experiment

3.1 Experimental Setup

Implementation Details. Training uses 8× NVIDIA H100 (80GB) GPUs, and real-world inference runs on a single RTX 5090 (32GB) GPU (More details in **Appendix A**).

Table 2: **Simulation Results on RoboTwin 2.0 Benchmark.** Comparison of task success rates. "†" indicates our reproduced results.

Method	Success Rate(↑)				
	Click B	Open L.	Place C.P.	Place B.S.	Avg.
RDT [33]	80.0%	59.0%	78.0%	5.0%	56.0%
ACT [70]	58.0%	56.0%	72.0%	7.0%	48.3%
DP [11]	54.0%	49.0%	41.0%	11.0%	38.8%
π_0 [2]	44.0%	85.0%	88.0%	23.0%	60.0%
OpenVLA-OFT† [20]	78.0%	66.0%	82.0%	9.0%	58.8%
ATI-VLA	**91.0%**	**87.0%**	**90.0%**	21.0%	**72.3%**

Table 3: **Efficiency Comparison Results.** All methods are evaluated on the same hardware. *Cost* denotes training time, with ATI-VLA including both Stage 1 and 2.

Method	**Latency↓**	**Throughput↑**	**Cost↓**
OpenVLA† [21]	0.203s	4.9Hz	28.9h
OpenVLA-OFT† [20]	0.105s	76.2Hz	30.2h
DreamVLA† [67]	0.111s	9.0Hz	33.5h
CoT-VLA† [69]	0.130s	7.7Hz	67.4h
ATI-VLA	0.107s	74.8Hz	16.2h
w/o side-path	0.105s	76.2Hz	-

Table 4: **Real-World Experiments on Different Platforms.** The table reports success rates at each stage for five long-horizon tasks, including an out-of-distribution (OOD) variant of T-shirt Folding. Results marked with "†" were reproduced using the same experimental settings as ATI-VLA. The average success rate corresponds to the completion rate of the entire task sequence.

Method	Object P		Markers C		Plate H			T-shirt F			T-shirt F(OOD)			Average
	Open	+Pack	C1	+C2	Pick	+Pass	+Place	S1	+S2	+S3	S1	+S2	+S3	
Galaxea R1 Lite platform														
OpenVLA-OFT†	18/25	17/25	17/25	15/25	18/25	17/25	15/25	17/25	15/25	14/25	10/25	8/25	8/25	55%
UniVLA†	14/25	13/25	13/25	13/25	15/25	14/25	12/25	15/25	15/25	12/25	13/25	10/25	7/25	46%
$\mathcal{F}_1$†	16/25	15/25	19/25	17/25	15/25	14/25	14/25	17/25	15/25	14/25	16/25	12/25	11/25	57%
DreamVLA†	15/25	14/25	17/25	14/25	13/25	11/25	11/25	15/25	14/25	14/25	13/25	12/25	10/25	50%
ATI-VLA(Ours)	**21/25**	**19/25**	**20/25**	**19/25**	**20/25**	**18/25**	**18/25**	**19/25**	**19/25**	**17/25**	**17/25**	**16/25**	**15/25**	**70%**
AgileX Cobot Magic platform														
OpenVLA-OFT†	16/25	15/25	19/25	16/25	17/25	17/25	15/25	16/25	15/25	15/25	9/25	8/25	7/25	54%
$\mathcal{F}_1$†	15/25	13/25	19/25	17/25	15/25	14/25	14/25	17/25	16/25	14/25	15/25	15/25	13/25	57%
ATI-VLA(Ours)	**21/25**	**20/25**	**21/25**	**19/25**	**19/25**	**18/25**	**18/25**	**20/25**	**19/25**	**18/25**	**18/25**	**17/25**	**15/25**	**72%**

Baselines. We compare ATI-VLA with representative state-of-the-art methods from two categories: **(1) General VLA models**, represented by OpenVLA-OFT [20] and π_0 [2]; and **(2) Predictive VLA models**, represented by $\mathcal{F}_1$ [35] and DreamVLA [67]. We further include an ablated variant, **ATI-VLA w/o alignment**, which jointly models future observations and actions in a single framework, omitting the explicit actionable representation alignment stage. For real-world experiments, we reproduce representative baselines (OpenVLA-OFT [20], UniVLA [5], DreamVLA [67], and $\mathcal{F}_1$ [35]) under the same training and deployment settings as ATI-VLA. More details in **Appendix B.3**.

Simulation & Real-World Experiments. We evaluate ATI-VLA on both simulation benchmarks and real-world robotic tasks. Simulation experiments are conducted on LIBERO [31] (Spatial, Object, Goal, and Long) and RoboTwin 2.0 [8]. For real-world evaluation, ATI-VLA is deployed on the Galaxea R1 Lite and AgileX Cobot Magic platforms and tested on five long-horizon tasks across four categories (Object Packing, Markers Collection, Plate Handover, and T-shirt Folding), including an additional out-of-distribution variant for T-shirt Folding. Each task includes 50 human-teleoperated demonstrations. Further details in **Appendix B.1** and **B.2**.

3.2 Overall Performance

Simulation Performance. As shown in Tab. 1, ATI-VLA achieves the best overall rank on LIBERO, outperforming both generalist and predictive VLA baselines, with strong results on the Object suite (99.4%) and Long suite (96.2%). On RoboTwin 2.0 (Tab. 2), ATI-VLA also obtains the highest average success rate, surpassing π_0 and OpenVLA-OFT, especially on fine-grained manipulation tasks such as Click Bell and Place Container Plate.

Real-World Performance. On Galaxea R1 Lite, ATI-VLA achieves a **70%** average success rate on long-horizon tasks (Tab. 4), outperforming DreamVLA (50%), $\mathcal{F}_1$ (57%), and OpenVLA-OFT (55%). It also performs well on OOD T-shirt folding (**60%**), demonstrating robust real-world generalization and reliable multi-stage control.

Efficiency Analysis. As shown in Tab. 3, ATI-VLA achieves the lowest training cost, requiring only 16.2h including both Stage 1 and Stage 2, much less than OpenVLA-OFT (30.2h), DreamVLA

Table 5: **Ablation Study on Latent Quality and Alignment.** Top: aligned latents outperform random noise. Bottom: alignment improves performance over unaligned representations.

SETTING	SPATIAL	OBJECT	GOAL	LONG
Injected Latent Quality				
RANDOM LATENT	95.6%	97.0%	94.4%	88.6%
ALIGNED LATENT	**98.6%**	**99.4%**	**97.2%**	**96.2%**
Effect of Representation Alignment				
ATI-VLA *w/o alignment*	96.4%	97.2%	95.8%	93.6%
ATI-VLA	**98.6%**	**99.4%**	**97.2%**	**96.2%**

Table 6: **Ablations on Injection Frequency.**

FREQUENCY	SPATIAL	OBJECT	GOAL	LONG
1	95.6%	97.8%	94.4%	92.8%
4	98.2%	98.8%	96.4%	93.2%
8	**98.6%**	**99.4%**	**97.2%**	**96.2%**
16	97.4%	99.0%	96.4%	93.0%

Table 7: **Quantitative Analysis on Actionable Representation Alignment.**

SETTING	CROSS-MODAL SIMILARITY ↑	CKNNA↑
BEFORE	0.31	0.2
AFTER	0.83	0.7

(33.5h), and CoT-VLA (67.4h). Its inference efficiency is only marginally lower than OpenVLA-OFT, while the *w/o side-path* variant shows that the side-path adds negligible overhead. Meanwhile, ATI-VLA substantially outperforms OpenVLA-OFT with nearly half the training time.

3.3 Ablation Studies

We perform ablation studies on the LIBERO benchmark to validate the effectiveness of our method.

Ablation on Actionable Representation Alignment Strategy. We evaluate the necessity of our shared codebook with joint decoding (Fig. 4(d)) by comparing three variants (Fig. 4(a)–(c)): a) shared codebook with dual decoders, b) shared codebook with sequential decoding, and c) dual codebooks with joint decoding. t-SNE visualizations demonstrate that only our proposed method achieves a well-mixed latent distribution, effectively bridging the observation–action modality gap. The quantitative results in Tab. 7 further confirm this trend, with improved cross-modal similarity and CKNNA after alignment. **Additional Stage-1 ablations in Appendix Tab. 14** further show the importance of action-side supervision and predictive alignment.

Ablation on Latent Representation Quality. To assess the role of injected latent quality, we compare ATI-VLA with variants that inject *randomly initialized latents* and remove explicit alignment (ATI-VLA *w/o alignment*) in Tab. 5. Injecting random latents substantially degrades performance, indicating that effective guidance requires semantically meaningful signals. Overall, these results suggest that explicit alignment is critical for producing predictive yet action-grounded priors.

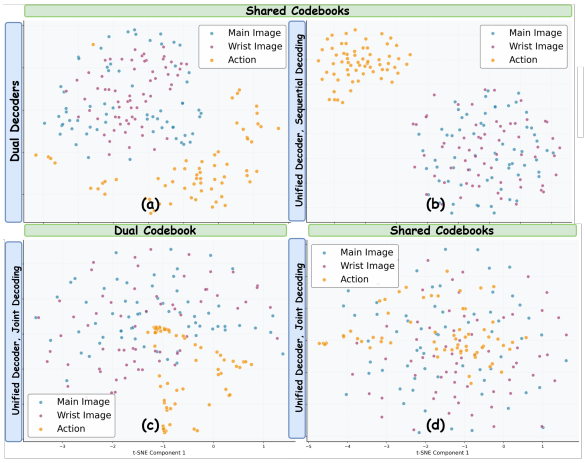
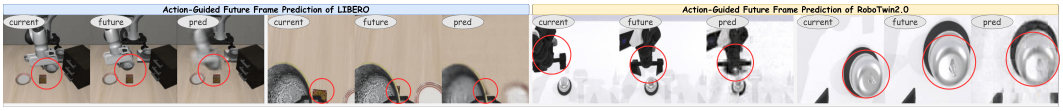


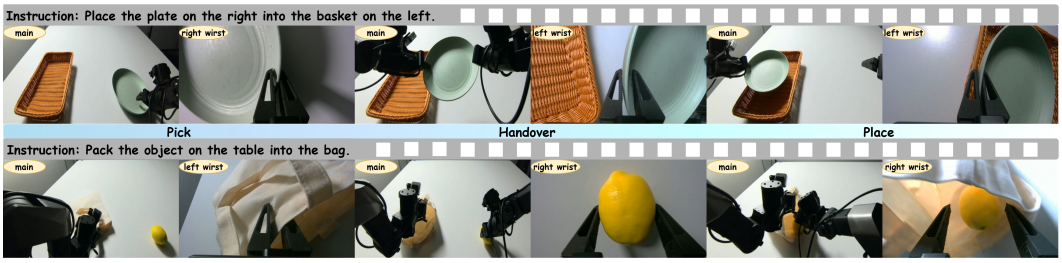
Figure 4: **Ablation on Actionable Representation Alignment Strategy.** t-SNE of observation/action latents under different alignment strategies: (a) dual decoder, (b) sequential decoding, (c) dual codebook, (d) our method (shared codebook + joint decoding).

Ablation on Frequency of Adaptive Injection. Tab. 6 ablates the injection frequency s. Dense injection ($s = 1$) hurts long-horizon performance, while overly sparse injection ($s = 16$) weakens predictive conditioning. The chosen setting $s = 8$ achieves the best trade-off between predictive guidance and action-decoding autonomy. Moreover, Tab. 9 shows that the preferred interval remains around one-third to one-fourth of the backbone depth, suggesting cross-backbone generalization with little tuning burden. **Appendix Tab. 13 further confirms this trend** across backbones with different architectural priors, including different attention and normalization schemes.

Ablation on Isolating the Source of Gains. We conduct controlled ablations to isolate the source of performance gains. Tab. 8 controls for backbone, training data, and evaluation protocol, showing



(a) Future Observation Prediction from Aligned Action Representations. Left: Current observation, ground-truth future, and predicted future (two views) on LIBERO. **Right:** RoboTwin 2.0 triplets. Red circles mark dynamic regions linked to imminent state transitions, suggesting the shared latent space captures actionable representations.



(b) Step-by-step Visualization of Two Real-World Tasks. Additional visualizations are in Appendix C.3.

Figure 5: Visualization results of the proposed method. (a) shows future observation prediction results. (b) illustrates step-by-step execution of real-world manipulation tasks.

Table 8: **Controlled Same-Backbone Ablation.** All variants share the same setup: (a) direct future prediction, (b) predictive injection without alignment, (c) alignment without adaptive injection, and (d) full ATI-VLA.

METHOD	LIBERO
DIRECT FUTURE PREDICTION	95.8%
INJECTION W/O ALIGNMENT	93.4%
ALIGNMENT W/O ADAPTIVE INJECTION	96.2%
ATI-VLA (OURS)	**97.9%**

Table 9: **Cross-Backbone Generalization.** The preferred injection interval s remains near one-third to one-fourth of model depth.

LLM	LAYERS L	BEST s
LLAMA-2-7B	32	8 (1/4)
QWEN2.5-1.5B	28	7 (1/4)
QWEN2.5-0.5B	24	8 (1/3)
QWEN2.5-3B	36	12 (1/3)

that the improvements mainly come from the full align-then-inject design. **We further control for different predicted modalities in Appendix Tab. 12**, confirming that the gains are not due to predicting a particular modality. Together, these results show that the gains mainly come from our align-then-inject design rather than other confounding factors.

3.4 Qualitative Analysis

The predicted observation frames in Fig. 5a, generated from aligned action representations, exhibit strong structural fidelity and temporal coherence with the ground truth, indicating that the shared latent space captures actionable scene dynamics. The real-world execution sequences in Fig. 5b further show reliable multi-stage manipulation, such as bag opening, inter-arm plate transfer, and precise placement, demonstrating the practical applicability of ATI-VLA. **Additional qualitative results are in Appendix C.3, with real-world videos in the supplementary material.**

The convergence curves in Fig. 6 show that ATI-VLA converges faster and achieves higher success rates than ATI-VLA w/o alignment, especially on long-horizon tasks. Compared with prior general and predictive VLA baselines, ATI-VLA also exhibits faster optimization and better final performance **(Fig. 1(f))**. These results suggest that actionable representation alignment provides stable predictive guidance and improves training efficiency.

4 Related Work

Vision–Language–Action Models. Vision–Language Models (VLMs) [1, 24, 55, 15, 12, 39, 54, 61, 43, 48, 57, 56, 72, 62, 63, 68, 42, 40] have substantially advanced robotic manipulation by tightly coupling visual perception with linguistic reasoning [53, 28, 66, 26, 41]. Building on this foundation, RT-2 [4] pioneered the integration of pre-trained VLMs into Vision–Language–Action modeling,

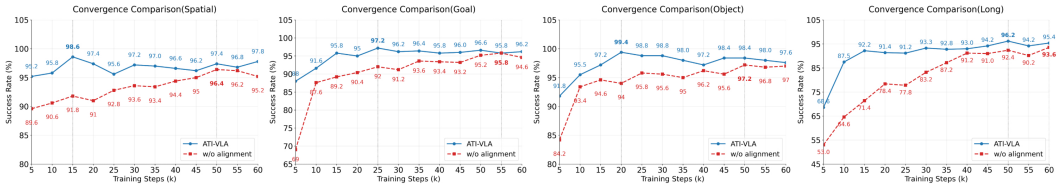


Figure 6: **Convergence Comparison.** ATI-VLA converges faster and performs better than ATI-VLA w/o alignment, while Fig. 1(f) further shows faster convergence over prior predictive VLA baselines.

while RT-X [46] further scaled this paradigm with large-scale data to improve generalization across tasks. A series of subsequent approaches, including OpenVLA [21], OpenVLA-OFT [20], and the π-series [2, 18] models, further adopt and extend this paradigm, demonstrating strong performance across a wide range of robotic manipulation tasks.

Predictive VLA Models. Recently, increasing attention has been paid to incorporating future prediction into Vision–Language–Action (VLA) models to enhance robotic manipulation [69, 67, 30, 65, 34, 17]. Some methods [14, 3] employ auxiliary image or video generation models to predict future frames and condition action prediction on the generated goals, introducing additional inference latency. Accordingly, subsequent methods aim to unify action prediction with future modeling, ranging from pixel-level forecasting [35, 71] to world knowledge prediction, within a single framework. Despite their strengths, these methods often suffer from modality misalignment and joint optimization conflicts that weaken action-centric learning. In contrast, ATI-VLA explicitly aligns predictive latents with actions and injects them under a single action-centric objective.

Robotic Manipulation in Learned Latent Spaces. Recent research explores compact, structured latent representations for robotic planning and control [58, 37, 22, 23]. VQ-BeT [22] and STAR [23] learn discrete action or skill representations to support transformer-based policy learning, while other approaches learn transferable latent skills from internet-scale human videos [60, 10, 5, 9, 38]. Both lines of work emphasize learning latent action or skill representations, whereas ATI-VLA targets the observation–action modality gap in predictive VLA. It also shares the use of discrete codebooks with recent action tokenizers [47, 13], but differs in the supervision and role of its codebook. Specifically, it learns shared, action-grounded prototypes by jointly predicting future observations and actions rather than reconstructing inputs. The resulting observation latents guide action decoding under a single action objective rather than serve as output action tokens. More closely related, VITA [36] also learns a shared visual–action codebook. ATI-VLA nevertheless differs in: (1) *codebook architecture and learning*, using a unified decoder for joint future observation–action prediction rather than separate autoencoders for modality-specific reconstruction; (2) *policy architecture*, injecting predictive latents through an adaptive side-path rather than generating shared tokens for dual visual–action decoding; and (3) *optimization*, freezing the alignment module and using only the action loss in Stage 2, whereas VITA [36] jointly optimizes observation and action losses during co-training.

5 Conclusion

We propose **ATI-VLA**, an action-centric predictive VLA framework that overcomes modality misalignment and optimization conflicts via an Actionable Alignment Then Adaptive Injection design. It first aligns observations and actions in a shared discrete latent space, then adaptively injects predictive latents under a single action objective, achieving state-of-the-art performance with faster convergence. This demonstrates how explicit alignment and adaptive injection can fully realize the potential of predictive VLA for robust robotic manipulation.

Acknowledgment

This study is supported by National Natural Science Foundation of China (Grant No. 62672144, 62636014, 62576076), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515010147, 2025B1515120017), Natural Science Foundation of Shenzhen City of China (Grant No. JCYJ20250604145700001), CCF-Tencent Rhino-Bird Open Research Fund, and Guangdong Research Team for Communication and Sensing Integrated with Intelligent Computing (Project No. 2024KCXTD047). The computational resources are supported by SongShan Lake HPC Center (SSL-HPC) in Great Bay University.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[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, 2024. *URL https://arxiv. org/abs/2410.24164*, 2024.

[3] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[5] 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.

[6] Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5817–5840. PMLR, 2024.

[7] 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.

[8] 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.

[9] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

[10] Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19752–19763, 2025.

[11] 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.

[12] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[13] Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, et al. Actioncodec: What makes for good action tokenizers. *arXiv preprint arXiv:2602.15397*, 2026.

[14] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[15] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Aaron J. Havens, Alexandre Araujo, Huan Zhang, and Bin Hu. Fine-grained local sensitivity analysis of standard dot-product self-attention. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17680–17696. PMLR, 2024.

[17] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[18] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[19] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pages 5562–5571. PMLR, 2021.

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

[21] 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.

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

[23] Hao Li, Qi Lv, Rui Shao, Xiang Deng, Yinchuan Li, Jianye Hao, and Liqiang Nie. Star: Learning diverse robot skill abstractions through rotation-augmented vector quantization. *arXiv preprint arXiv:2506.03863*, 2025.

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[25] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[26] Wei Li, Jizhihui Liu, Li Yixing, Junwen Tong, Rui Shao, and Liqiang Nie. Consisvla-4d: Advancing spatiotemporal consistency in efficient 3d-perception and 4d-reasoning for robotic manipulation. *arXiv preprint arXiv:2605.05126*, 2026.

[27] Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint arXiv:2508.21046*, 2025.

[28] Zaijing Li, Bing Hu, Rui Shao, Gongwei Chen, Dongmei Jiang, Pengwei Xie, Jianye Hao, and Liqiang Nie. Global prior meets local consistency: Dual-memory augmented vision-language-action model for efficient robotic manipulation. *arXiv preprint arXiv:2602.20200*, 2026.

[29] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[30] Minghui Lin, Pengxiang Ding, Shu Wang, Zifeng Zhuang, Yang Liu, Xinyang Tong, Wenxuan Song, Shangke Lyu, Siteng Huang, and Donglin Wang. Hif-vla: Hindsight, insight and foresight through motion representation for vision-language-action models. *arXiv preprint arXiv:2512.09928*, 2025.

[31] 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.

[32] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[33] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[34] Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. Last _{0}: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026.

[35] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025.

[36] Xiangkai Ma, Lekai Xing, Han Zhang, Wenzhong Li, and Sanglu Lu. Unifying perception and action: A hybrid-modality pipeline with implicit visual chain-of-thought for robotic action generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22380–22390, 2026.

[37] Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control, 2024. *URL https://arxiv. org/abs/2407.15840*.

[38] Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

[39] Yuxin Peng, Zishuo Wang, Geng Li, Xiangtian Zheng, Sibo Yin, and Hulingxiao He. A survey on fine-grained multimodal large language models. *Chinese Journal of Electronics*, 35(2):771–803, 2026.

[40] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10015–10023. IEEE, 2019.

[41] Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025.

[42] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and grounding multi-modal media manipulation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913. IEEE, 2023.

[43] Rui Shao, Tianxing Wu, Jianlong Wu, Liqiang Nie, and Ziwei Liu. Detecting and grounding multi-modal media manipulation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5556–5574, 2024.

[44] Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Zhijun Li, Donglin Wang, Lujia Wang, et al. Pd-vla: Accelerating vision-language-action model integrated with action chunking via parallel decoding. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13162–13169. IEEE, 2025.

[45] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024.

[46] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[47] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11089–11099. IEEE, 2025.

[48] Yelin Wang, Zijia Song, Shuo Ye, Chuanguang Yang, Miaoyu Wang, Yong Xu, Zhulin An, Yongjun Xu, and Zitong Yu. Rsicclm: A multimodal large language model for remote sensing image change captioning. In *European Conference on Computer Vision*, pages 253–272. Springer, 2026.

[49] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025.

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

[51] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

[52] Junjie Wen, Yichen Zhu, Minjie Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression. In *Forty-second International Conference on Machine Learning*, 2025.

[53] Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: Improving vision-language-action models with active visual attention. *arXiv preprint arXiv:2511.18960*, 2025.

[54] Linhui Xiao, Guiping Cao, Mingyue Guo, Xianchao Guan, Fan Yang, Ming Tao, Xin Li, Yuxin Peng, and Yaowei Wang. A survey on the green development of large models: From resource-efficient architectures to hardware–software co-design. *Chinese Journal of Electronics*, 35(5):1817–1840, 2026.

[55] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

[56] Hailong Yan, Junjian Huang, and Tingwen Huang. Igdnet: Zero-shot robust underexposed image enhancement via illumination-guided and denoising. *IEEE Transactions on Artificial Intelligence*, 2025.

[57] Hailong Yan, Shice Liu, Tao Wang, Xiangtao Zhang, Yijie Zhong, Jinwei Chen, Le Zhang, and Bo Li. Animeagent: Is the multi-agent via image-to-video models a good disney storytelling artist? *arXiv preprint arXiv:2602.20664*, 2026.

[58] Chenyu Yang, Davide Liconti, and Robert K Katzschmann. Vq-ace: Efficient policy search for dexterous robotic manipulation via action chunking embedding. *arXiv preprint arXiv:2411.03556*, 2024.

[59] Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. Efficientvla: Training-free acceleration and compression for vision-language-action models. *arXiv preprint arXiv:2506.10100*, 2025.

[60] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[61] Zhaoda Ye, Xiangteng He, and Yuxin Peng. Unsupervised cross-media hashing learning via knowledge graph. *Chinese Journal of Electronics*, 31(6):1081–1091, 2022.

[62] Kaishen Yuan, Zitong Yu, Xin Liu, Weicheng Xie, Huanjing Yue, and Jingyu Yang. Auformer: Vision transformers are parameter-efficient facial action unit detectors. In *European Conference on Computer Vision*, pages 427–445. Springer, 2024.

[63] Kaishen Yuan, Zitong Yu, Xin Liu, Bohao Xing, Yuting Zhang, Weicheng Xie, Linlin Shen, and Björn W Schuller. Multi-granularity facial emotional representation with unlabeled data and textual supervision. *IEEE Transactions on Image Processing*, 35:3479–3494, 2026.

[64] Nikolay Yudin, Sergei Kudriashov, Alexander Gaponov, and Maxim Rakhuba. Pay attention to attention distribution: A new local lipschitz bound for transformers. *arXiv preprint arXiv:2507.07814*, 2025.

[65] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

[66] Likui Zhang, Tao Tang, Zhihao Zhan, Xiuwei Chen, Zisheng Chen, Jianhua Han, Jiangtong Zhu, Pei Xu, Hang Xu, Hefeng Wu, et al. Atomicvla: Unlocking the potential of atomic skill learning in robots. *arXiv preprint arXiv:2603.07648*, 2026.

[67] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025.

[68] Ganlong Zhao, Guanbin Li, and Yizhou Yu. Navgemini: a multi-modal llm agent for vision-and-language navigation. *Visual Intelligence*, 4(1):1, 2026.

[69] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[70] 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.

[71] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

[72] Yijie Zhu, Yibo Lyu, Zitong Yu, Rui Shao, Kaiyang Zhou, and Liqiang Nie. Emosym: A symbiotic framework for unified emotional understanding and generation via latent reasoning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 5451–5460, 2025.

[73] Yijie Zhu, Rui Shao, Ziyang Liu, Jie He, Jizhihui Liu, Jiuru Wang, and Zitong Yu. H-gar: A hierarchical interaction framework via goal-driven observation-action refinement for robotic manipulation. *arXiv preprint arXiv:2511.17079*, 2025.

ATI-VLA: Action-Centric Predictive Vision–Language–Action Models via Actionable Alignment Then Adaptive Injection

Appendix

This appendix supplements the main paper with implementation details, additional experimental results, and theoretical analyses. It is organized as follows:

- Section A details the model architectures and training configurations used in simulation and real-world experiments.
- Section B describes the benchmarks, real-world tasks, evaluation protocols, and baselines.
- Section C presents extended ablation studies, analyses of observation–action modality gaps, optimization conflicts, and failure modes, as well as additional qualitative results.
- Section D provides theoretical analyses of the shared codebook and interval-based latent injection.
- Sections E and F discuss the limitations of ATI-VLA, future research directions, and broader societal implications, including potential deployment risks.

A Implementation Details.

A.1 Model Details

Structural Details of Actionable Representation Alignment. In the Actionable Representation Alignment stage of ATI-VLA, we implement three core components:

- **Observation Encoder** employs a patch-embedding layer followed by a spatial Transformer to extract visual tokens, which are then quantized into a shared discrete latent space via a unified codebook. It processes a single input frame and outputs a set of discrete latent codes that capture spatial and semantic information of the scene.
- **Action Encoder** uses a lightweight MLP to project the current action vector into a spatially-aligned embedding grid, which is similarly quantized through the same shared codebook. This ensures that action representations share the same discrete latent structure as visual observations, enabling cross-modal alignment.
- **Unified Decoder** is built with stacked self-attention and cross-attention layers, taking the quantized latent representations of both observation and action as context to jointly predict the future observation and future action. The self-attention captures intra-modal dependencies, while cross-attention fuses the aligned observation and action latents for coherent multi-modal prediction.

Base Framework and Implementation. ATI-VLA adopts OpenVLA-OFT [20] as its implementation framework, with OpenVLA-7B [21] serving as the underlying backbone. We retain the standard OpenVLA-OFT implementation setup and incorporate the proposed actionable representation alignment and adaptive latent injection mechanisms.

Inference Mechanism. Unlike video-generation-based policies, ATI-VLA does not require explicit future-observation decoding during inference. Consistent with the use of predictive latent representations in DreamVLA [67] and Seer [45], ATI-VLA directly exploits predictive information in latent space to guide action generation. Specifically, in stage 1, the alignment module is trained with both future-observation and action supervision to learn future-aware, action-grounded representations. This module is subsequently frozen but retained at inference time: the current observation is mapped into the learned latent space, and the resulting predictive latent is adaptively injected into the action decoder as an explicit guidance signal. Therefore, although ATI-VLA does not perform an explicit future-observation rollout at test time, the predictive information learned in Stage 1 continues to directly guide action generation through the retained adaptive side-path.

A.2 Training Details

Simulation Training Setup. For the LIBERO benchmark, the network is adapted using Low-Rank Adaptation (LoRA), configured with a rank of 32 and a scaling parameter $\alpha = 64$. Training proceeds with a batch size of 64, initiated at a learning rate of 5×10^{-4}. The action sequence length is set to $n = 8$. Validation is performed on benchmark tasks at intervals of every 5,000 steps, with the model checkpoint delivering the highest validation performance being selected for final reporting. Regarding the RoboTwin 2.0 benchmark, we utilize a batch size of 32, an initial learning rate of 5×10^{-4}, and an action chunk size of 25.

Real-World Training Setup. For real-world bimanual manipulation, we follow the standard OpenVLA-OFT setup, using one main-camera view and two wrist-camera views with an action chunk length of $n = 25$. The model is fine-tuned with LoRA, maintaining a rank of 32 and $\alpha = 64$. The training regimen spans 80,000 steps with a batch size of 32. The learning rate is initially set to 5×10^{-4} and is subsequently decayed to 5×10^{-5} after 50,000 steps. Starting from the 60,000th step, model checkpoints are evaluated every 5,000 steps. The checkpoint exhibiting the best performance is retained as the final model for deployment.

B Benchmarks and Baselines

B.1 Simulation Benchmark

LIBERO is a language-conditioned manipulation benchmark that evaluates complementary facets of robotic capability through multiple task suites.

- LIBERO-Spatial contains 10 tasks with identical objects arranged in varied layouts, emphasizing spatial reasoning.
- LIBERO-Object includes 10 tasks with fixed layouts but different object instances, testing object-level generalization.
- LIBERO-Goal comprises 10 tasks sharing object categories and layouts while varying goals, assessing goal-conditioned behavior diversity.
- LIBERO-Long consists of 10 challenging long-horizon tasks across diverse objects and scenes, stressing temporal reasoning and sequential control.

We follow the standard protocol and use 50 expert demonstrations per task from the official release.

RoboTwin 2.0 provides a large-scale benchmark for bimanual manipulation, built on an asset library of 731 objects across 147 categories. It defines 50 dual-arm tasks with expert trajectories generated via an MLLM-driven simulation pipeline. To enhance robustness and sim-to-real transfer, it incorporates structured domain randomization across clutter, lighting, backgrounds, table height, and instructions. For evaluation, four tasks are selected: **Click Bell, Open Laptop, Place Container Plate, and Place Bread Skillet**, each evaluated over 100 trials.

B.2 Real-World Tasks

Task Descriptions. We deploy ATI-VLA on Galaxea R1 Lite and AgileX Cobot Magic platforms for real-world evaluation across five long-horizon sequential manipulation tasks that span four core capability categories: Object Packing (evaluating dual-arm coordination and container manipulation), Markers Collection (testing fine-grained precision manipulation), Plate Handover (assessing dynamic object transfer and bimanual interaction), and T-shirt Folding (examining deformable object handling in both standard and OOD configurations). These tasks collectively challenge fundamental robotic capabilities including precision manipulation, bimanual coordination, bimanual handover, and dynamic adaptation under uncertainty. Tasks requiring dual-arm coordination (Object Packing, Plate Handover, and both T-shirt Folding variants) evaluate spatial-temporal synchronization and inter-arm transfer, while Markers Collection isolates fine motor control in a single-arm precision scenario. Each task is evaluated through progressive stage success rates to assess intermediate performance, with 50 human-teleoperated demonstrations collected per task to ensure sufficient training data diversity. Details are provided below.

- **Object Packing**
  Instruction:"Pack the object on the table into the bag."
  This dual-arm sequential manipulation task evaluates coordinated bimanual operation for container handling and object packing, requiring precise temporal coordination and spatial reasoning. The progressive stages include: 1) *"Open"* – securely open the bag and maintain its accessible configuration, and 2) *"+Pack"* – accurately placing tabletop item into the bag while maintaining bag stability.
- **Markers Collection**
  Instruction:"Put the markers into the cup."
  This task assesses the limits of fine-grained precision manipulation through slender object handling and constrained placement, demanding high precision and delicate handling. The two-stage operation involves: 1) *"C1"* – picking up the first marker and inserting it into the cup with precise orientation control, and 2) *"+C2"* – repeating the process for the second marker while avoiding collisions with the first.
- **Plate Handover**
  Instruction:"Place the plate on the right into the basket on the left."
  This dual-arm handover task evaluates dynamic object transfer and inter-arm coordination, focusing on seamless spatial-temporal synchronization during object exchange. The three-stage sequence comprises: 1) *"Pick"* – grasping the target plate securely with the right arm, 2) *"+Pass"* – smoothly transferring the plate to the left arm through a coordinated handover maneuver that maintains object stability throughout the exchange, and 3) *"+Place"* – accurately positioning the plate onto the left-side target plate.
- **T-shirt Folding (In-distribution)**
  Instruction:"Fold the T-shirt."
  This deformable object manipulation task examines fabric handling and structured folding execution, demanding comprehensive multi-step planning and sustained task coherence across complex operation chains. Following the established three-stage protocol: 1) *"Step 1"* – firmly grasping the T-shirt edge and executing the first controlled folding motion across the garment surface, 2) *"Step 2"* – repeating the folding motion to reinforce the fold, and 3) *"Step 3"* – grasping the right side of the T-shirt and folding it over to the left side.
- **T-shirt Folding (Out-of-Distribution Variant)**
  Instruction:"Fold the T-shirt."
  This out-of-distribution variant evaluates generalization capabilities when confronted with systematic variations in garment attributes, while preserving the core folding skill. The task follows the identical three-stage folding procedure, but introduces key variations in the T-shirt properties: 1) the garment exhibits differing visual characteristics including color not seen during training, and 2) the T-shirt dimensions and sizes are altered.

Data Collection and Augmentation. Real-world training data for all tasks is collected via human teleoperation, with 50 expert demonstrations per task. To enhance generalization, moderate data augmentation—varying initial object poses, spatial arrangements, and environmental configurations—is systematically applied to Object Packing, Markers Collection, Plate Handover, and the out-of-distribution T-shirt Folding variant. The in-distribution T-shirt folding task maintains consistent fabric type and initial layout to establish a baseline for deformable object manipulation. Augmentation strategies include randomized object positions, container orientations, and initial arm configurations to simulate realistic execution variability.

B.3 Baselines

We compare ATI-VLA against representative state-of-the-art methods from two categories.

1) General VLA models. These methods directly map the current observation and language instruction to actions, without explicit future prediction:

- OpenVLA-OFT [20] is an efficiency-optimized variant of OpenVLA that improves inference speed while maintaining strong general-purpose manipulation performance.
- π_0 [2] follows a dual-system design that combines high-level reasoning with low-level control for long-horizon tasks.

Table 10: **Trade-off Between Model Complexity and Task Performance.**

Method	Parameters	FLOPs	LIBERO	RoboTwin 2.0	Real World	Cost (h)
OpenVLA-OFT (Baseline)†	7.71 B	8.45 T	96.6	58.8	55.0	30.2
ATI-VLA	**8.04 B** (+4.2%)	**8.51 T** (+0.7%)	**97.9** (+1.3)	**72.3** (+13.5)	**70.0** (+15.0)	**16.2** (-46.4%)

Table 11: **Controlled Same-Backbone Ablation.** All variants use the same backbone, the same training data, and the same evaluation protocol.

Method	LIBERO	RoboTwin 2.0	Real-World
Baseline + Direct Future Prediction	95.8%	52.4%	48%
Baseline + injection without alignment	93.4%	51.3%	50%
Baseline + with alignment but without adaptive injection	96.2%	60.7%	62%
ATI-VLA (Ours)	**97.9%**	**72.3%**	**70%**

- CogVLA [27] is a Cognition-Aligned Vision-Language-Action framework that leverages instruction-driven routing and sparsification to improve both efficiency and performance.
- PD-VLA [44] is a performance/efficiency-oriented baseline that accelerates action decoding, serving as a strong non-predictive reference.

2) Predictive VLA models. These methods incorporate look-ahead signals (e.g., future observations, latent dynamics, or world states) to guide action generation:

- Seer [45] learns predictive inverse dynamics by forecasting future visual states and predicting actions conditioned on the forecast, enabling scalable pretraining and sim-to-real adaptation.
- DreamVLA [67] forecasts compact world knowledge (dynamic/spatial/semantic cues) and uses it for inverse dynamics in a perception–prediction–action loop, with structured attention to disentangle signals and a diffusion-based Transformer for action modeling.
- CoT-VLA [69] adds explicit visual chain-of-thought by autoregressively predicting future frames as intermediate visual goals, then generating short action chunks to reach these goals.
- $\mathcal{F}_1$ [35] integrates visual foresight into decision making via a modular Mixture-of-Transformers, predicting goal-conditioned future visuals as planning targets and generating actions through foresight-guided inverse dynamics.
- H-GAR [73] introduces a hierarchical interaction framework that refines observation–action trajectories in a goal-driven manner for robotic manipulation.
- UniVLA [5] learns cross-embodiment VLA policies by deriving task-centric latent action representations from videos, enabling scalable pretraining on heterogeneous (including human) videos and efficient deployment via latent action decoding.

We further include an ablated variant, **ATI-VLA w/o alignment**, which jointly models future observations and actions in a single framework but removes the explicit Actionable Representation Alignment stage, isolating the effect of alignment from other components of the framework.

C Extended Analyses

C.1 Extended Ablations

Trade-off Analysis. As shown in Table 10, ATI-VLA consistently improves task performance over OpenVLA-OFT with only 0.7% additional FLOPs, while reducing total training time by 46.4% (30.2 to 16.2 hours). These results demonstrate that ATI-VLA improves policy performance while maintaining comparable computational complexity and substantially reducing training cost.

Controlled Same-Backbone Ablation. To isolate where the performance gains come from, we conduct a controlled same-backbone ablation. All variants use the same backbone, training data, and evaluation protocol, and differ only in their predictive design. As shown in Table 11, direct

Table 12: **Controlled Comparison over Predicted Modalities.** All variants use the same backbone, training data, and evaluation protocol, differing only in the predicted modality and predictive design. Results report average success rates.

Method	LIBERO	RoboTwin 2.0	Real-World
Image Prediction (CoT-VLA-like)	95.2%	52.0%	50%
World-Knowledge Prediction (DreamVLA-like)	94.4%	50.3%	46%
ATI-VLA (Ours)	**97.9%**	**72.3%**	**70%**

Table 13: **Generalization of Injection Interval Across Backbones.** We evaluate LLM backbones with different attention patterns and normalization schemes. The preferred injection interval remains around one-third to one-fourth of the model depth.

LLM	Architectural Prior	Norm	Layers	Best s
LLaMA-2-7B	Dense causal attention	RMSNorm	32	8 $(1/4)$
Mistral-7B	Sliding-window attention	RMSNorm	32	8 $(1/4)$
OPT-1.3B	Dense causal attention	LayerNorm	24	8 $(1/3)$

future prediction yields only limited improvement, while injection without alignment even degrades performance, suggesting that unaligned predictive features may interfere with action decoding. Incorporating alignment improves performance, and the full ATI-VLA achieves the best results across both simulation and real-world settings. These results indicate that the gains primarily stem from the combination of actionable alignment and adaptive injection, rather than from extra supervision, model capacity, or training budget.

Controlled Comparison over Predicted Modalities. To account for different predicted modalities for a fair comparison, we reproduce two representative variants under the same framework: an image-prediction variant following the spirit of CoT-VLA [69], and a world-knowledge-prediction variant following the spirit of DreamVLA [67]. As shown in Table 12, both variants underperform ATI-VLA, especially on RoboTwin 2.0 and real-world tasks. This suggests that the gains do not simply come from predicting a particular modality, but from making predictive representations actionable through alignment and injecting them into action decoding in an adaptive, action-centric manner.

Generalization of Injection Interval Across Backbones. We conduct ablation experiments on the injection interval using LLM backbones with different attention patterns and normalization schemes. The results in Tab. 13 show that the best interval remains around one-third to one-fourth of the model depth, indicating that this trend is not specific to specific backbone. Overall, dense injection tends to over-constrain action decoding, while overly sparse injection provides insufficient predictive guidance. Thus, interval-based injection reflects a general trade-off, although the exact optimum may slightly vary across backbone families. Importantly, this does not introduce a substantial tuning burden, since the empirical search space is largely narrowed to a small range around one-third to one-fourth of the backbone depth.

Ablation on Stage-1 Alignment Design. We further analyze the design of Stage-1 alignment in Table 14. Although Stage 2 injects observation-side latents, action information in Stage 1 is still essential because the goal is to learn an action-grounded shared latent space. Compared with observation-only alignment, using both observations and actions improves performance. This indicates that action-side supervision helps observation latents encode control-relevant factors rather than merely visual appearance. We also compare predictive alignment with a simpler reconstruction-based objective. Reconstruction-based alignment achieves lower performance across all settings. This suggests that reconstruction alone is insufficient to learn future-oriented actionable semantics. In contrast, our predictive alignment objective encourages the shared latent space to capture action-induced state transitions, leading to stronger and more transferable action guidance.

Statistical Significance. We report the mean and standard deviation of ATI-VLA across random seeds on LIBERO. ATI-VLA achieves a 97.9% average success rate with only 0.1% standard deviation, indicating stable performance across seeds. The small variance on all suites further suggests that the reported gains are not due to random fluctuations.

Table 14: **Ablation on Stage-1 Alignment Design.** We examine whether action information and predictive learning are necessary for learning action-grounded latent representations.

Setting	LIBERO	RoboTwin 2.0	Real-World
Effect of Action Information in Stage 1			
Observation-only Alignment	95.6%	64.3%	61%
Full Alignment (Ours)	**97.9%**	**72.3%**	**70%**
Effect of Predictive Alignment Objective			
Reconstruction-based Alignment	96.0%	62.4%	59%
Predictive Alignment (Ours)	**97.9%**	**72.3%**	**70%**

Table 15: **Failure-Mode Diagnosis in Long-Horizon Real-World Tasks.** We categorize failures on Plate Handover and T-shirt Folding, with 100 trials per task and 200 trials in total for each method. The non-aligned variant fails more often in semantic grounding, pose/control grounding, and transition-phase stages.

Method	Semantic	Pose/Control	Transition	Success
w/o Alignment	26	33	49	92
ATI-VLA (Ours)	**11**	**19**	**41**	**129**

C.2 Deeper Analysis of Modality Gaps and Optimization Conflicts.

C.2.1 Modality Gaps.

(1) What kind of failures occur. As shown in Table 15, the non-aligned variant does not fail diffusely; instead, its failures concentrate on concrete and interpretable cases, including wrong target selection, inaccurate approach poses, mis-oriented placements, and failures to switch actions after subgoal transitions. We categorize these instance-level errors into semantic grounding, pose/control grounding, and transition-phase failures. These failure modes directly reflect the need for strong modality alignment, since the policy must translate predictive semantics into the correct next action chunk. This is particularly critical in Plate Handover and T-shirt Folding, where successful execution requires repeated semantic-to-action remapping across multiple subgoals.

(2) When the first dominant failure occurs. As shown in Table 16, the first dominant failures of the non-aligned variant are concentrated in later or transition-heavy stages, such as *Pass*/*Place* in Plate Handover and the later folding stages in T-shirt Folding. These stages require the policy to re-ground updated predictive semantics into a new control regime after completing previous subgoals. In contrast, ATI-VLA reduces failures across all stages and achieves higher final success rates, suggesting that explicit alignment improves the reliability of semantic-to-action remapping in long-horizon execution.

(3) Whether alignment is lower immediately before failure. As shown in Table 17, using cross-modal similarity as the alignment metric, the non-aligned variant exhibits a clear drop from safe steps to pre-failure steps, with a gap of 0.22. In contrast, ATI-VLA maintains much higher similarity on both safe and pre-failure steps, and reduces the gap to 0.10. This indicates that failures are closely associated with weakened observation–action alignment, especially in high-risk stages immediately before execution errors. Overall, these analyses show that modality gaps lead to identifiable failures at key stages, including wrong target selection, incorrect approach or placement poses, and subgoal-switching errors. Such failures are especially frequent in multi-stage transition states, where predictive semantics must be converted into a new action regime. ATI-VLA effectively mitigates these failures by maintaining stronger action-grounded predictive representations.

C.2.2 Optimization Conflicts.

We further examine whether joint prediction-action training introduces optimization conflicts. Specifically, we compute the gradient cosine similarity between the prediction loss and the action loss on shared parameters. As shown in Table 18, both DreamVLA and our direct-future-prediction variant exhibit negative average cosine similarity and a high ratio of negative gradient pairs, indicating that the two objectives often provide conflicting update directions. Applying PCGrad partially reduces this conflict, improving the average cosine similarity from -0.29 to -0.09 and increasing performance

Table 16: **First Dominant Failure Stage in Long-Horizon Real-World Tasks.** We report the stage where the first dominant failure occurs on Plate Handover and T-shirt Folding, with 100 trials per task. The non-aligned variant fails more often in later or transition-heavy stages.

Plate Handover

Method	Pick	Pass	Place	Success
w/o Alignment	8	18	29	45
ATI-VLA (Ours)	**5**	**11**	**22**	**62**

T-shirt Folding

Method	S1	S2	S3	Success
w/o Alignment	10	15	28	47
ATI-VLA (Ours)	**4**	**9**	**20**	**67**

Table 17: **Cross-Modal Similarity Before Failure.** We compare cross-modal similarity on safe steps and pre-failure steps. The non-aligned variant exhibits a larger alignment drop before failure, while ATI-VLA maintains higher alignment in these high-risk regions.

Method	Safe Steps ↑	Pre-Failure Steps ↑	Gap ↓
w/o Alignment	0.59	0.37	0.22
ATI-VLA (Ours)	**0.80**	**0.70**	**0.10**

on both RoboTwin 2.0 and real-world tasks. This confirms that gradient interference is a real optimization issue. In contrast, ATI-VLA avoids such conflict by design: after Stage-1 alignment, Stage-2 injection is optimized solely with the action loss, leading to the best performance.

C.2.3 Why Is the Modality Gap Severe in Predictive VLA Settings.

Predictive VLA makes the observation–action modality gap more severe because the predicted representation is often learned under an observation-oriented objective but later used for action generation. Specifically, future-observation prediction encourages visually accurate reconstruction of the next scene. However, many visual details, such as background texture, illumination, or pixel-level appearance, are not directly relevant to control. As a result, the predictive latent may encode rich visual information while failing to emphasize action-relevant factors such as target identity, contact state, object pose, or subgoal transition. When such observation-oriented latents are used to guide the action decoder, the policy must implicitly translate them into executable action chunks, which enlarges the modality gap.

This issue is further amplified by joint prediction-action optimization. In conventional predictive VLA training, the shared representation is optimized by both the prediction loss and the action loss. These two objectives can prefer different features: the prediction loss favors visually faithful future states, whereas the action loss favors compact and control-relevant abstractions. As analyzed in our gradient-conflict study, their gradients can point in conflicting directions, pulling the shared representation away from a purely action-centric solution. Therefore, the modality gap in predictive VLA is not only caused by the inherent difference between observations and actions, but also by the mismatch between the prediction objective and the downstream control objective.

ATI-VLA addresses this issue through the align-then-inject design. Stage 1 uses both observation and action information to learn an action-grounded shared latent space, so that predictive latents encode future-oriented but control-relevant semantics. Stage 2 then injects these aligned latents into action decoding under a single action objective, avoiding further prediction-action optimization conflict. This makes predictive information more directly usable for control rather than leaving the policy to bridge the modality gap implicitly.

C.3 Additional Qualitative Results

Figure 7 presents extended qualitative results of future observation prediction generated from aligned action representations on both the LIBERO and RoboTwin 2.0 benchmarks. Across diverse scene layouts, object configurations, and task types, the predicted future frames consistently maintain structural fidelity and temporal coherence with the ground truth. Notably, the predictions preserve fine-grained object details and spatial relationships even in cluttered environments and under varying camera viewpoints. This indicates that the aligned latent space successfully encodes actionable scene dynamics that are both predictive and semantically grounded, effectively bridging the observation-

Table 18: **Gradient Conflict Between Prediction and Action Objectives.** We measure the gradient cosine similarity between prediction and action losses on shared parameters. PCGrad partially alleviates the conflict, while ATI-VLA avoids it by design and achieves the best performance.

Method	Avg. Cos$(g_{\text{pred}}, g_{\text{act}})$ ↑	Neg. Ratio ↓	RoboTwin 2.0 ↑	Real-World ↑
DreamVLA	-0.23	43%	–	50%
Direct Future Prediction	-0.29	39%	52.4%	48%
Direct Future Prediction + PCGrad	-0.09	21%	57.6%	55%
ATI-VLA (Align + Inject)	**Avoided by design**	**Avoided by design**	**72.3%**	**70%**

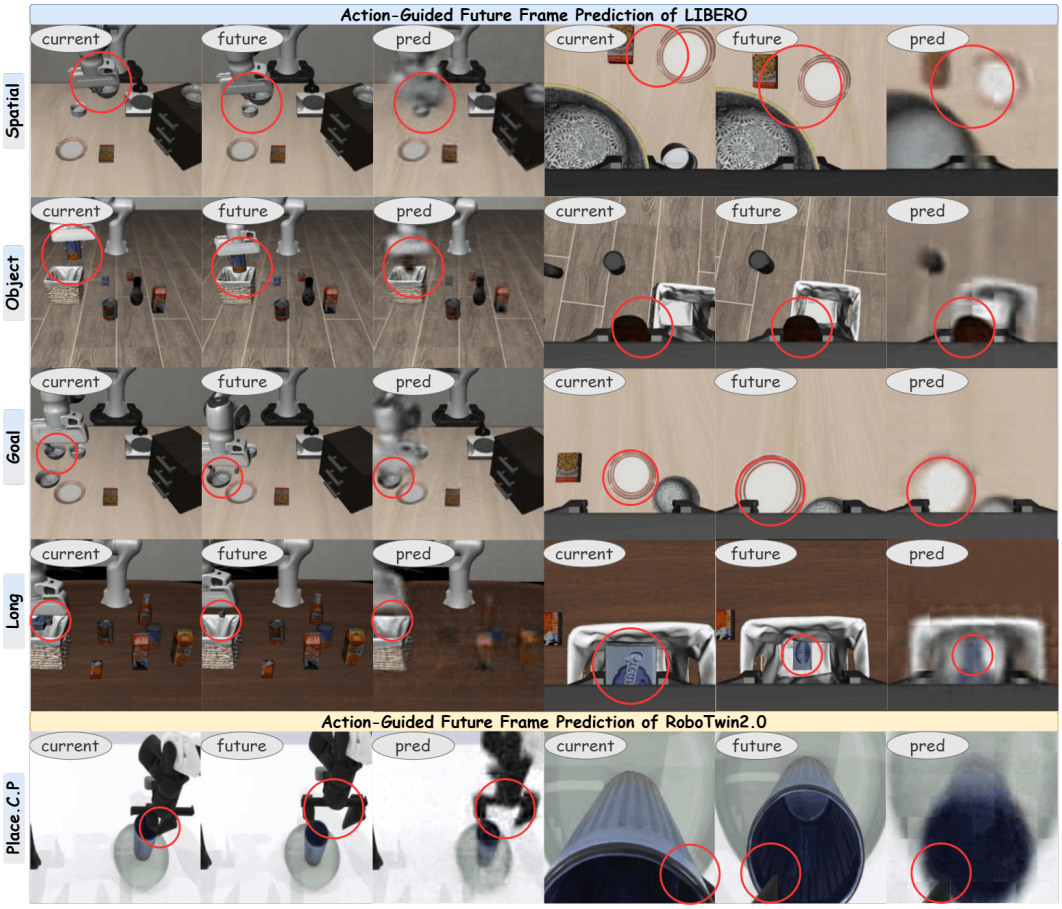


Figure 7: **Extended Future Observation Prediction from Aligned Action Representations.** Additional qualitative examples of future observation prediction on the LIBERO and RoboTwin 2.0 benchmarks. Each triplet shows the current observation, ground-truth future observation, and the predicted future observation from the aligned action representation.

action modality gap. The robustness of these predictions across different task suites further confirms that the alignment strategy yields generalizable representations that transfer well beyond the training distribution.

Figure 8 illustrates additional real-world execution sequences of all tasks. In the **Object Packing** sequence, the model demonstrates reliable dual-arm coordination—first opening the bag with stable bimanual manipulation, then precisely placing the object inside while maintaining bag posture. The **Markers Collection** sequence highlights fine-grained precision manipulation as the robot picks up slender markers and inserts them into a cup with controlled orientation. The **Plate Handover** sequence shows smooth inter-arm transfer: securely grasping the plate with the right arm, performing a stable handover to the left arm, and accurately placing it onto the target. The **T-shirt Folding** sequences (both standard and OOD variants) demonstrate robust deformable object manipulation, maintaining coherent folding motions across varying fabric properties and initial configurations.



Figure 8: **Extended Step-by-Step Real-World Task Execution.** Further visualization of all real-world tasks. Each frame captures a critical stage in the sequential execution.

These sequences collectively highlight the framework's capacity for temporally extended planning, robust bimanual synchronization, and adaptive control under real-world uncertainties.

Together, these extended visualizations reinforce that ATI-VLA's *Align-Then-Inject* design not only produces semantically consistent future predictions but also translates these predictions into stable, precise, and generalizable real-world manipulation behavior. The alignment stage ensures that predictive representations are grounded in the action space, while the adaptive injection mechanism integrates this guidance in a manner that preserves the autonomy and precision of action decoding.

D Theoretical Grounding

D.1 Shared Discrete Codebook

Let the observation and action representations be

$$z_O = f_O(O_t), \qquad z_A = f_A(A_t),$$

and let the shared codebook be

$$\mathcal{C} = \{c_k\}_{k=1}^{K}, \quad c_k \in \mathbb{R}^d.$$

The quantization operator is defined as

$$Q(z) = c_{k^*(z)}, \quad k^*(z) = \arg\min_k \|z - c_k\|_2^2,$$

yielding shared latent representations

$$\tilde{z}_O = Q(z_O), \qquad \tilde{z}_A = Q(z_A).$$

We focus on the excess action risk induced when predictive observation latents are used for action generation. Let $\pi(\cdot)$ denote the action decoder and $\ell(\pi(z), a)$ the loss. Assuming $\ell(\pi(z), a)$ is L-Lipschitz with respect to z, we obtain:

$$\begin{aligned} \Delta\mathcal{R} &:= |\mathbb{E}[\ell(\pi(\tilde{z}_O), a)] - \mathbb{E}[\ell(\pi(\tilde{z}_A), a)]| \\ &\leq \mathbb{E}\left[|\ell(\pi(\tilde{z}_O), a) - \ell(\pi(\tilde{z}_A), a)|\right] \\ &\leq L \cdot \mathbb{E}\|\tilde{z}_O - \tilde{z}_A\|_2. \end{aligned} \tag{16}$$

Equation (16) shows that the excess action risk is upper-bounded by the latent discrepancy between predictive and action representations. Therefore, the role of the shared codebook is to explicitly reduce

$$\mathbb{E}\|\tilde{z}_O - \tilde{z}_A\|_2,$$

thereby directly tightening the action risk bound.

Furthermore, the cross-modal discrepancy can be decomposed as

$$\|z_O - z_A\|_2 \le \|z_O - \tilde{z}_O\|_2 + \|\tilde{z}_O - \tilde{z}_A\|_2 + \|\tilde{z}_A - z_A\|_2, \tag{17}$$

where the middle term $\|\tilde{z}_O - \tilde{z}_A\|_2$ represents the discrepancy in a finite prototype space. In particular, if

$$Q(z_O) = Q(z_A),$$

then

$$\tilde{z}_O = \tilde{z}_A \;\Rightarrow\; \Delta\mathcal{R} = 0.$$

Additionally, the codebook is learned under the joint objective

$$\mathcal{L}_{\text{align}} = \lambda_O \mathcal{L}_{\text{obs}} + \lambda_A \mathcal{L}_{\text{act}}, \tag{18}$$

which enforces both future observation prediction and future action prediction. Hence, the learned prototypes satisfy

$$\tilde{z}_O, \tilde{z}_A \in \mathcal{C}, \quad \min \mathbb{E}\|\tilde{z}_O - \tilde{z}_A\|_2, \quad \text{s.t. } \mathcal{L}_{\text{obs}} + \mathcal{L}_{\text{act}} \text{ jointly minimized.} \tag{19}$$

This explains why the shared discrete codebook reduces the modality gap.

D.2 Interval Injection

We analyze interval injection as a local first-order perturbation of the action-token decoding trajectory. This local view is important because standard dot-product self-attention is not globally Lipschitz on an unbounded input domain [19]. Instead of assuming global Lipschitzness of the full Transformer, we only require bounded first-order sensitivity in the neighborhood actually traversed by the clean and injected action-token states, which is aligned with recent local analyses of attention and Transformer blocks [6, 16, 64].

Let Φ_l denote the layer-l transition restricted to the action-query substate, and let Ξ_l collect the remaining contextual variables, including observation, language, non-action tokens, normalization statistics, and residual context. For the clean and injected action-token states $\widehat{H}_l^{\text{act}}$ and $\widetilde{H}_l^{\text{act}}$, define

$$E_l := \widetilde{H}_l^{\text{act}} - \widehat{H}_l^{\text{act}}. \tag{20}$$

We use the following local stability assumption.

Assumption (Local Jacobian bound). For each layer l, there exists a neighborhood $\mathcal{N}_l$ containing the clean and injected action-token states such that

$$\sup_{U \in \mathcal{N}_l} \|\partial_H \Phi_l(U, \Xi_l)\|_2 \le \rho_l. \tag{21}$$

This assumption does not require the full Transformer to be globally Lipschitz. It only bounds the local Jacobian of the action-token transition along the trajectories relevant to ATI-VLA Step 2.

Let the injection layers be

$$\mathcal{I}_s := \{l : \; l \bmod s = 0\}. \tag{22}$$

At an injection layer l, the predictive side-path contributes

$$\Delta_l := \beta_l \,\text{CrossAttn}\Big(\Phi_l(\widetilde{H}_{l-1}^{\text{act}}, \Xi_l), Z_{\mathcal{O}_t}^{(l)}, Z_{\mathcal{O}_t}^{(l)}\Big). \tag{23}$$

The clean and injected action-token substates then evolve as

$$\widehat{H}_l^{\text{act}} = \Phi_l(\widehat{H}_{l-1}^{\text{act}}, \Xi_l), \qquad \widetilde{H}_l^{\text{act}} = \Phi_l(\widetilde{H}_{l-1}^{\text{act}}, \Xi_l) + \Delta_l \,\mathbf{1}_{\{l \in \mathcal{I}_s\}}. \tag{24}$$

By the mean value theorem and the local Jacobian bound,

$$\|E_l\|_2 \le \rho_l \|E_{l-1}\|_2 + \|\Delta_l\|_2 \,\mathbf{1}_{\{l \in \mathcal{I}_s\}}. \tag{25}$$

Unrolling this recursion yields

$$\|E_L\|_2 \leq \sum_{l \in \mathcal{I}_s} \left(\prod_{k=l+1}^{L} \rho_k \right) \|\Delta_l\|_2. \tag{26}$$

Equation (26) shows that each injected predictive perturbation is propagated through the remaining action-decoding layers. Since $|\mathcal{I}_s| \approx L/s$, dense injection introduces more perturbation sources and can over-condition the action-token trajectory, whereas overly sparse injection provides too few predictive signals to guide decoding effectively. Thus, the injection interval reflects a trade-off between predictive guidance and perturbation accumulation:

$$\mathcal{E}_{\text{lack}}(s) \uparrow \text{ with } s, \qquad \mathcal{E}_{\text{perturb}}(s) \downarrow \text{ with } s. \tag{27}$$

Therefore, this analysis should be interpreted as a local, trend-level perturbation argument for the adaptive injection stage, rather than as a tight global bound or an exact prediction of the optimal interval. It supports the existence of an intermediate injection regime: denser layer-wise injection accumulates predictive perturbations more strongly through depth, while overly sparse injection weakens predictive guidance. This theoretical trend is consistent with our empirical findings in Table 13, where the preferred interval remains around $L/3$ to $L/4$ across different LLM backbones.

E Limitations and Future Work

Although ATI-VLA shows strong performance on simulation benchmarks and real-world robotic tasks, the current training and evaluation are still based on a relatively limited set of manipulation datasets and task scenarios. In particular, the real-world experiments mainly rely on task-specific teleoperated demonstrations, which validate the effectiveness of our align-then-inject design but do not fully cover the diversity of objects, scenes, embodiments, and interaction patterns in open-world manipulation. A natural direction for future work is to scale ATI-VLA to larger and more heterogeneous robot datasets. Such data could help the alignment module learn a more comprehensive action-grounded latent space and improve the transferability of predictive latents across tasks and platforms. Future work may also incorporate human videos, cross-embodiment trajectories, and self-supervised interaction data to provide richer priors about object affordances and action-induced state transitions, further extending ATI-VLA toward open-ended long-horizon manipulation.

F Broader Impact and Potential Risks

ATI-VLA aims to improve the reliability and efficiency of VLA models for long-horizon robotic manipulation. By grounding predictive representations in action generation, it may benefit applications such as assistive robotics, warehouse automation, and flexible manufacturing, while also offering a general direction for action-centric predictive learning in embodied AI.

Potential risks may arise when stronger robotic policies are deployed in unconstrained or safety-critical environments, where prediction errors, distribution shifts, or incorrect instruction grounding could lead to unintended behaviors. Practical deployment should therefore include safety constraints, human oversight, permission control, and thorough real-world validation. Future work should further study uncertainty-aware execution and failure detection for safer open-world operation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our contributions in the abstract and at the end of the introduction, while the scope is outlined in the abstract and at the beginning of the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mentioned our limitations in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete (and correct) proof in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide implementation details in Appendix to ensure the reproducibility of the experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We train and evaluate our method using open-sourced datasets and models, which facilitates easy replication. Following acceptance, both the code and the data will be made available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We detail our experimental setting in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report experimental results under different random seed settings in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We introduce this in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not engage in research that violates ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We describe the potential impacts in Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We describe the potential impacts in Appendix.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We utilized open-source models and datasets in accordance with the license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We develop comprehensive guidelines for running our code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used solely for writing and grammatical refinement purposes.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.