3A-VLA: Abstraction-Aligned Action Learning for Vision-Language Agents in 3D Game Worlds

Zheyuan Zhou[1,2*] **Liang Du**[1,*] **Zixun Sun**[1,2,*,†]
Xiaoyu Zhou[1] **Ruimin Ye**[1] **Qihao Chen**[1] **Yinda Chen**[1] **Lemiao Qiu**[2]

[1]IEG, Tencent [2]Zhejiang University

https://3a-vla.github.io/

Abstract

Despite advances in vision-language-action (VLA) models, they remain limited in highly dynamic game settings such as 3D open worlds and competitive player versus player (PvP) settings, where agents must efficiently extract sparse, actionable signals from dense visual input while integrating multimodal cues to track off-screen high-value targets in real time. To address this, we introduce **3A-VLA** (**A**bstraction-**A**ligned **A**ction VLA), a framework that grounds action learning in explicit intention and environment abstractions rather than superficial pattern matching. We introduce dual task-agnostic abstractions: the intention abstraction (IA), which condenses verbose instructions and reasoning into explicit semantic primitives, and the environment semantics abstraction (ESA), which structures dense visual streams into a spatial–functional affordance representation to guide grounded actions. We further propose an abstraction alignment reweighting (AAR) module that adaptively reweights the action imitation loss. A continuous intention–environment alignment signal is used to emphasize reliable action supervision when abstractions agree and reduce the influence of ambiguous demonstrations when they diverge, thereby learning a policy that balances fine-grained control and high-level reasoning without the need for manual rules. Extensive experiments show that 3A-VLA performs competitively among the evaluated baselines in both open-world (Minecraft) and competitive PvP (Peacekeeper Elite) settings. It also demonstrates zero-shot combat transfer from Peacekeeper Elite to high-fidelity games, including Valorant, CS2, GTA V, Elden Ring, and Mount & Blade.

1 Introduction

Recent vision-language-action (VLA) models have enabled embodied agents to interpret multimodal instructions and execute complex tasks within immersive digital environments [2, 31, 33]. These methods have demonstrated strong performance in structured, relatively static embodied settings [6, 42, 16] and structured yet varied interactions in PvE game environments [9, 28]. However, in dynamic game environments defined by completely spontaneous, real-time interactions, existing models still face a fundamental challenge: perceptual overload [21]. To navigate an open world with emergent events (e.g., suddenly appearing vehicles) or a competitive player versus player (PvP) match with sudden threats (e.g., abruptly emerging opponents), the model must continuously process complex sensory data while parsing instructions that are often verbose. Existing approaches lack the capacity to efficiently exploit the most essential signals for decision-making from this large amount of data with these signals remaining implicitly embedded and difficult to access without explicit selection mechanisms [24]. This deficiency is rooted in a profound modality misalignment between continuous visual features and discrete language tokens [25, 38, 19, 42]. Owing to this substantial

*Equal contribution. † Corresponding author.

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

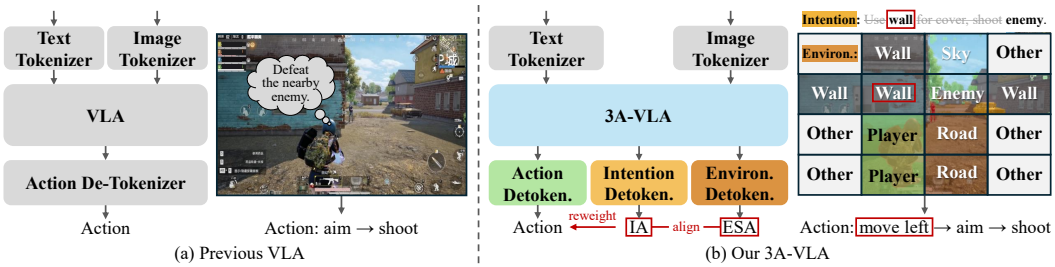


Figure 1: Unlike previous VLA models which mapped low-level inputs directly to actions, 3A-VLA aligns action learning with intention abstraction (IA) and environment semantics abstraction (ESA) through multimodal detokenizers. As shown in a PvP scenario, 3A-VLA grounds instructions and visual scenes into high-level, interpretable concepts such as "wall" and "enemy". **(i)** When trained on action sequences such as "move left, aim, shoot," the model learns the semantic relationship between "wall" and "move left," recognizing the object's functional role as tactical cover rather than merely its presence. **(ii)** When tasked with locating off-screen enemies, the model uses the structured affordance representation from the ESA to reason about likely ambush points, such as inferring that a "wall" may serve as high-probability cover from which an opponent could emerge, thereby enabling proactive searches. In off-screen scenarios with weak IA-ESA correlation, semantic reasoning outweighs action prediction; 3A-VLA adapts by reweighting action learning to prioritize semantic grounding.

semantic gap, existing models struggle to align high-level instructions with low-level observations. This forces agents to rely on superficial pattern matching, often referred to as "instruction overfitting" or "causal confusion", rather than achieving true cross-modal understanding. Ultimately, this results in fragile policies that fail to generalize, leading to unreliable decision-making in complex, open-world environments [17, 11]. Open-world data show 2–3×higher intra-action visual diversity than tabletop domains like LIBERO, with more varied observations per action (Appendix I.2).

To bridge this gap, we draw inspiration from the cognitive and neural sciences, which reveal that the human brain does not process all sensory data equally [30]. Instead, as shown in Figure 2, it relies on a conscious bottleneck, that filters vast streams of irrelevant information and abstracts high-level semantics to extract only actionable primitives. This process allows the brain to ignore the noise of detailed textures and wordy sentences and instead focus on sparse, abstract representations of reality. We posit that for embodied agents to move beyond the following of simple instructions to truly understand intent, they must incorporate a similar inductive bias: learning to actively abstract the hidden, intention-derived information from redundant multimodal data streams. Guided by this principle, we introduce **3A-VLA** (**A**bstraction-**A**ligned **A**ction VLA). Unlike standard VLA models that passively map sensory inputs to actions, 3A-VLA trains the model to align action learning with task-relevant abstractions from noisy multimodal data streams. We instantiate this framework via a unified causal Transformer [32] that acts as a cognition-inspired bottleneck, synthesizing two complementary abstraction pathways. First, through the intention abstraction (IA), the model aligns its internal representations with compact intention primitives (e.g., the concept of a "wall" in the text in Figure 1). This process involves extracting the latent strategic intent from verbose instructions, encouraging the agent to internalize a high-level task structure rather than relying on superficial phrase matching. Simultaneously, to mitigate visual clutter, we employ the environment semantics abstraction (ESA). By projecting visual streams into a sparse topological representation of task-critical affordances (e.g., the entity "wall" in the semantic map in Figure 1), our method prioritizes the functional structure of the environment over irrelevant textural details. Both pathways are supervised by zero-shot off-the-shelf components (an open-source VLM for IA keywords and a pretrained segmenter for the ESA), with no retrieval, domain-specific fine-tuning, human annotation, or more powerful annotator model needed. As a benefit of the abstraction procedure, the model's internal attention naturally concentrates on task-relevant regions, which enables a parameter-free token-pruning strategy that trims visual tokens at inference with negligible performance loss.

We further introduce the abstraction alignment loss reweighting (AAR) module, enabling an adaptive policy that ties action supervision to high-level semantic alignment. As shown in Figure 1, this governs situational dynamics ranging from in-view execution to off-screen search. Specifically, AAR reweights the action imitation loss using a continuous alignment signal derived from the intention and environment abstractions. Rather than switching modes, the model emphasizes action

supervision when alignment is strong (e.g., target in view), while reducing the influence of ambiguous demonstrations as alignment weakens (e.g., target absent, the scene is task-agnostic). This soft reweighting mechanism allows the policy to remain guided by semantic abstractions without the need for manual intervention. Our contributions are summarized as follows:

- We propose **3A-VLA**, an **A**bstraction-**A**ligned **A**ction VLA framework that grounds decision-making in the alignment between the agent intention and the environment semantics, thereby increasing robustness over pattern-matching baselines in dynamic gaming environments.
- We propose a dual-channel abstraction to achieve semantic grounding. The intention abstraction (IA) compresses verbose instructions into actionable semantic primitives, whereas the environment semantics abstraction (ESA) aligns perceptual representations with the functional topology of the environment by encoding spatial affordances. Together, they form a coherent abstract state that enables robust situational reasoning.
- We propose the abstraction alignment reweighting (AAR) module, which adaptively weights action imitation via a continuous intention–environment alignment signal, allowing the policy to emphasize reliable demonstrations while down-weighting ambiguous ones.
- Extensive experiments demonstrate the state-of-the-art performance of 3A-VLA on open-world (Minecraft) and competitive PvP (Peacekeeper Elite) benchmarks, as well as its zero-shot combat transfer (trained on Peacekeeper Elite) to Valorant, CS2, GTA V, Elden Ring, Mount & Blade, and Minecraft.

2 Related work

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models ground visual and linguistic inputs into actions. RT-1 [6] and RT-2 [42] established large-scale vision-language-action learning, while OpenVLA [16] and Octo [12] further improved generalization through scalable transformer architectures. Diffusion Policy [10] and the π-series [4, 3] introduced generative action modeling for precise control. Despite these advances, existing approaches often directly fuse multimodal inputs without explicitly separating task-relevant intention from environmental information, making them vulnerable to irrelevant details. Embodied Chain-of-Thought (ECoT) [37] introduces intermediate reasoning but primarily operates at the token level. In contrast, we introduce structured intention and environment abstractions to guide action learning.

2.2 Information Selection and Reweighting

Reducing redundant information has been explored at both inference and training stages. FastV [8] prunes visual tokens based on attention scores, while ToMe [5] merges redundant tokens to accelerate inference. Meanwhile, L2RW [26], Meta-Weight-Net [27], learned data manipulation [15], and Co-teaching [14] improve training through adaptive sample weighting or selection. Our approach integrates task-aware abstractions with connectivity-based token selection and relation-conditioned action-loss reweighting, emphasizing task-relevant information for efficient and robust action learning.

2.3 Embodied Agents in Digital Environments

Digital games provide complex and reproducible environments for embodied agents. Existing approaches broadly fall into API-based and pixel-based paradigms. API-based agents, such as Voyager [31] and Ghost in the Minecraft [41], leverage LLMs for high-level planning but often rely on privileged interfaces. Pixel-based agents, including VPT [2] and STEVE-1 [20], learn visuomotor control from raw observations but remain limited in semantic planning and visual robustness. CombatVLA [9] extends VLA methods to 3D combat, while Cradle [29], JARVIS-VLA [18], and OpenHA [34] explore more general agentic frameworks. However, real-time control and robustness remain challenging. Our 3A-VLA bridges these paradigms by integrating high-level semantic abstractions with pixel-based control in a unified architecture.

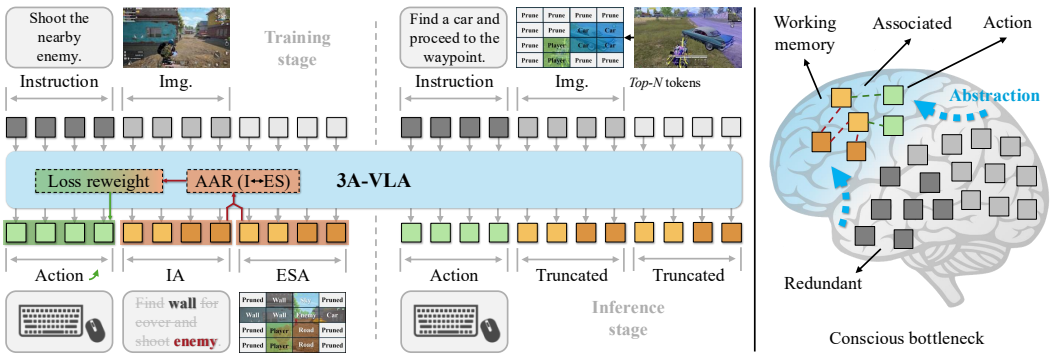


Figure 2: **Overview.** 3A-VLA learns abstraction-aligned actions by coupling intention abstraction (IA) and environment semantics abstraction (ESA) with abstraction alignment reweighting (AAR). During training, IA and ESA expose sparse task primitives from language and vision, while AAR reweights action supervision according to intention–environment alignment. At inference, the induced sparsity enables low-latency control by keeping only top-N task-critical visual tokens.

3 Methodology

3.1 Overview

As illustrated in Figure 2, the agent receives an observation $\mathbf{x}_v$ and a language instruction $\mathbf{x}_l$; let $\mathbf{X} = (\mathbf{x}_v, \mathbf{x}_l)$, and let $\mathbf{A}$ denote the expert action. In dynamic environments, direct imitation learning strategies face two practical difficulties: (i) observations are dense and often dominated by task-irrelevant details such as texture, background motion. (ii) short-horizon action labels are often ambiguous. In particular, the same action may be demonstrated under different intents and scene contexts, while similar observations may require different actions depending on the task. As an abstraction-aligned action learner, 3A-VLA augments a standard autoregressive VLA with two compact abstraction targets: an intention target $\mathbf{y}_{\text{ia}}$ and an environment semantic target $\mathbf{y}_{\text{esa}}$. Specifically, $\mathbf{y}_{\text{ia}} \in \mathcal{V}_{\text{ia}}^{L_{\text{ia}}}$ is a short sequence of intent keywords, and $\mathbf{y}_{\text{esa}}$ is a flattened coarse semantic grid over tactical categories. The IA summarizes the agent's goal and strategy, whereas the ESA summarizes which task-relevant information is present in the visual scene. Both targets serve solely as auxiliary training signals and are omitted at inference. All the components are implemented with a single autoregressive causal Transformer θ initialized from Qwen2-VL [32]. The same backbone is used to predict expert actions with IA/ESA targets during training. During inference, the policy follows the standard VLA deployment pipeline and uses only the actions from the predictions.

3.2 Intention abstraction (IA)

Linguistic instructions in open-ended worlds are verbose and unstructured (e.g., "*Find a way to quickly enter the safe zone to escape the hazard zone*"), yet the policy-relevant intent is sparse and discrete (e.g., `[waypoint, safezone]`). The IA provides a compact intent summary as $\mathbf{y}_{\text{ia}} \in \mathcal{V}_{\text{ia}}^{L_{\text{ia}}}$.

Zero-shot intent generation. Standard trajectory datasets do not provide intent labels. We therefore synthesize IA targets offline using a single off-the-shelf open-source VLM, Qwen2-VL-7B [32]. In contrast to prior works [9, 28] that rely on verbose chain-of-thought (CoT) rationales for supervision, we leverage the VLM's structured output from a single inference pass: while the model generates both reasoning steps and conclusions, we retain only the final decisive keyword sequence for supervision. We detail the annotator prompt design and the complete keyword enumeration in Appendix F.

$\mathbf{y}_{\text{ia}}$ encourages the shared representation to preserve task intent in a compact and stable form. Unlike free-form rationales and chain-of-thought annotations, the keyword supervision reduces redundancy and exposes the policy to reusable task primitives. Consequently, this compact semantic representation aligns naturally with ESA, enabling a unified latent space where linguistic intent and visual scene structure are coupled. Appendix F explores the scaling potential of our method by replacing off-the-shelf components with advanced alternatives, including retrieval-augmented generation (RAG) systems and stronger VLMs.

3.3 Environment semantics abstraction (ESA)

Pixel streams in embodied control are massively redundant; the important information for many short-horizon decisions is the topology of the task-relevant affordances over the per-pixel RGB values. The ESA provides a compact scene-level semantic target $\mathbf{y}_{\mathrm{esa}}$ that summarizes the functional structure of the current observation.

Latent semantic grid. We compress the image into an $H_g \times W_g$ semantic grid $\mathbf{G}_{\mathrm{sem}}$. Ground-truth grids are obtained by running an off-the-shelf segmenter (SegFormer-B2 [35]) on $\mathbf{x}_v$ and projecting the dense mask into five coarse categories: `person`, `vehicle`, `cover`, `item`, and `other`. No domain-specific fine-tuning is performed. Within each patch $\Omega_{u,v}$, we keep the top-K classes under a tactical priority $\rho(\texttt{person}) > \rho(\texttt{vehicle}) > \rho(\texttt{cover}) > \rho(\texttt{item}) > \rho(\texttt{other})$:

$$\mathbf{g}_{u,v} \;=\; \mathrm{Top}_K\big(\{c \mid c \in \Omega_{u,v}\},\, \rho\big). \tag{1}$$

This priority-aware pooling strategy ensures that small but important foreground entities, such as distant people, are retained even when they occupy only a small fraction of a grid cell. We flatten $\mathbf{G}_{\mathrm{sem}}$ into an ordered token sequence $\mathbf{y}_{\mathrm{esa}}$ via a raster scan, with the K class tokens inside each cell emitted in the order of their priority. The ordered-tuple convention provides a consistent tokenization for each grid cell and preserves the priority among cooccurring tactical categories. $\mathbf{y}_{\mathrm{esa}}$ encourages the shared representation to focus on the functional scene structure rather than the raw texture information. The coarse grid format is intentionally sparse and task oriented; denser alternatives such as whole-image reconstruction and dense semantic prediction are less effective, as demonstrated by ablation studies in Table 3 and Table 17.

3.4 Abstraction alignment reweighting (AAR)

The IA and ESA describe complementary abstractions of each demonstration: IA specifies the intended objective, while ESA captures the relevant environmental affordances. The AAR reweights the action imitation loss according to the semantic alignment between these two abstractions.

Let $\mathbf{H}_i \in \mathbb{R}^{T_i \times d}$ be the final-layer hidden states for sample i. For $a \in \{\mathrm{ia}, \mathrm{esa}\}$, let $\mathcal{T}_i^a$ denote the token span of the IA or ESA. We obtain span-level representations by mean pooling:

$$\mathbf{h}_i^a = \frac{1}{|\mathcal{T}_i^a|} \sum_{t \in \mathcal{T}_i^a} \mathbf{H}_{i,t}, \quad a \in \{\mathrm{ia}, \mathrm{esa}\}, \tag{2}$$

where both $\mathbf{h}_i^{\mathrm{ia}}$ and $\mathbf{h}_i^{\mathrm{esa}}$ lie in $\mathbb{R}^d$. We construct the IA–ESA relation feature as

$$\mathbf{u}_i^a = \frac{\mathcal{D}(\mathbf{h}_i^a)}{\|\mathcal{D}(\mathbf{h}_i^a)\|_2 + \epsilon}, \quad a \in \{\mathrm{ia}, \mathrm{esa}\}, \qquad \mathbf{r}_i = \mathbf{u}_i^{\mathrm{ia}} \odot \mathbf{u}_i^{\mathrm{esa}}. \tag{3}$$

The stop gradient operator $\mathcal{D}$ prevents gradients in the reweighting branch from directly updating the hidden states. $\odot$ is the Hadamard product. The element-wise product preserves dimension-wise IA–ESA interactions for the scorer.

An MLP-based scorer g_ψ is employed to estimate the importance of each relation feature. Specifically, g_ψ first projects the relation feature $\mathbf{r}_i \in \mathbb{R}^d$ into a lower-dimensional bottleneck space, and then processes it using two residual MLP layers with layer normalization. A linear projection head maps the encoded relation feature to a scalar score. The resulting scores are normalized across the batch via a softmax operation to yield batch-relative action weights (Appendix N provides details of g_ψ.):

$$q_i = g_\psi(\mathbf{r}_i), \qquad p_i = \frac{\exp(q_i)}{\sum_{j=1}^{B} \exp(q_j)}, \qquad w_i = B\, p_i, \tag{4}$$

where the scaling factor B guarantees that $\frac{1}{B}\sum_{i=1}^{B} w_i = 1$, thereby preserving the overall magnitude of the action supervision signal. To avoid degenerate concentration on a few samples, we regularize the normalized entropy $\bar{H}$ of the batch-relative distribution:

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} \Big[w_i \mathcal{L}_{\mathrm{act}}^{(i)} + \mathcal{L}_{\mathrm{abs}}^{(i)} \Big] - \lambda_{\bar{H}} \bar{H}(\mathbf{p}), \qquad \bar{H}(\mathbf{p}) = \frac{-\sum_{i=1}^{B} p_i \log(p_i + \epsilon)}{\log B}, \tag{5}$$

where $\mathcal{L}_{\mathrm{act}}^{(i)}$ is the action imitation loss, and $\mathcal{L}_{\mathrm{abs}}^{(i)}$ consists of the token-level IA and ESA losses. We fix $\lambda_{\bar{H}} = 0.01$ for all experiments without task-specific tuning. Appendix H provides hyperparameter sensitivity and robustness experiments of $\lambda_{\bar{H}}$.

Table 1: **Evaluation on Minecraft.** Tasks are categorized into "Embodied", "Combat", and "GUI". We report the average steps ("Steps") and the success rate ("SR", %). 3A-VLA (full model): IA (Qwen2-VL-7B, zero-shot), ESA (SegFormer-B2, zero-shot), AAR. '-' denotes task failure. Bold indicates best performance. Stronger annotator variants are reported in Appendices F and G.

Model	Size	Embodied Tasks			Combat Tasks			GUI Tasks		
		Steps ↓	SR (Mini) ↑	SR (All) ↑	Steps ↓	SR (Mini) ↑	SR (All) ↑	Steps ↓	SR (Mini) ↑	SR (All) ↑
Previous Methods										
VPT [2]	248M	377	$10.1^{\pm 2.8}$	$6.0^{\pm 1.4}$	396	$3.6^{\pm 1.5}$	$3.6^{\pm 1.2}$	398	$0.7^{\pm 0.1}$	$0.8^{\pm 0.3}$
STEVE-1 [20]	248M	384	$8.4^{\pm 2.4}$	$8.0^{\pm 1.8}$	395	$4.9^{\pm 1.4}$	$3.9^{\pm 1.1}$	391	0.0	$3.2^{\pm 1.3}$
ROCKET-1 [7]	72B	392	$19.2^{\pm 3.5}$	$18.9^{\pm 3.6}$	320	$29.8^{\pm 4.2}$	$27.9^{\pm 4.8}$	-	-	-
Vision-Language-Action Models										
JARVIS-VLA [18]	7B	305	$31.0^{\pm 4.9}$	$30.0^{\pm 5.3}$	352	$18.3^{\pm 3.1}$	$18.5^{\pm 3.6}$	339	$25.3^{\pm 4.2}$	$25.1^{\pm 4.5}$
OpenHA [34]	7B	287	$37.0^{\pm 5.8}$	$30.1^{\pm 4.9}$	316	$40.0^{\pm 6.1}$	$31.9^{\pm 5.2}$	314	$33.3^{\pm 5.0}$	$32.5^{\pm 5.3}$
3A-VLA (Ours)	7B	**278**	$\mathbf{40.6}^{\pm 3.2}$	$\mathbf{33.5}^{\pm 2.6}$	**271**	$\mathbf{45.9}^{\pm 3.5}$	$\mathbf{35.4}^{\pm 2.9}$	**307**	$\mathbf{34.7}^{\pm 2.7}$	$\mathbf{34.8}^{\pm 2.6}$

4 Experiments

4.1 Experimental setup

Benchmarks. For **Minecraft**, we follow OpenHA [34] and evaluate on MCU [40], comprising 800+ embodied, combat, and GUI tasks. We report Mini (10 representative tasks per domain) and All (800+ tasks) protocols in out-of-distribution environments with novel seeds. **Peacekeeper Elite** (PE) covers six atomic tasks at intermediate difficulty (gold/silver tiers): precision parachuting, resource scavenging, combat engagement, teammate revival, vehicle acquisition, and strategic rotation. For **cross-game combat transfer**, PE-combat-trained models are evaluated without target-game fine-tuning in Valorant, CS2, GTA V, Elden Ring, Mount & Blade, and Minecraft (melee and ranged combat). Hit SR requires at least one valid strike on a dynamic target; Kill SR requires defeating the designated target. Table 5 reports Hit SR; Appendix K.1 provides complete Hit/Kill results and protocols. These metrics assess short-horizon combat rather than general game completion.

Metrics. We employ three primary metrics to assess performance and efficiency: *Success Rate (SR):* A task is considered successful if the goal is achieved. *Steps:* Average environmental steps per task completion, serving as a proxy for execution efficiency. *Latency:* The mean input-to-action time (s) per step.

Training and evaluation settings. We fine-tune Qwen2-VL-7B [32] independently for each environment. For Minecraft, the training set consists of approximately 50 hours of gameplay data, whereas the Peacekeeper Elite model is trained on a separate 200-hour corpus. In each dataset, the task instructions, our proposed IA and ESA data, and ground-truth frame-action pairs are aligned. Unless otherwise noted, the primary benchmark results use the *simplest* instantiation: IA keywords were generated in a zero-shot manner by Qwen2-VL-7B, and ESA semantic maps were produced by the pretrained SegFormer-B2 [35] without any domain-specific fine-tuning. Stronger annotators and fine-tuned segmenters are analyzed in Section 4.5 and Appendices F and G; pruning is a separate deployment study in Section 4.6. Training is distributed across 8 NVIDIA H20 GPUs, while inference is performed on a single L40S GPU. The agent operates strictly on first-person RGB frames (640×360) and maps outputs to a discretized human-like action space. For ESA, we set the grid resolution to $H_g \times W_g = 8 \times 8$ with $K = 2$, where detailed sensitivity analysis and label noise robustness are deferred to Appendices F, G, and J.2 (Tables 10, 11, and 17). External-baseline versions, paired-manifest evaluation, run-to-run uncertainty, and MCU Mini aggregation are detailed in Appendix N.2.

Comparison methods. Minecraft baselines are VPT [2], STEVE-1 [20], ROCKET-1 [7], JARVIS-VLA [18], and OpenHA [34]. For Peacekeeper Elite, we compare proprietary foundation models using zero-shot CoT, Qwen2-VL-7B, controlled Qwen2-VL-7B IL variants trained on identical data, and the CombatVLA reproduction and adapted Open-P2P listed in Table 2. Controlled variants use CoT/caption supervision as IA/ESA alternatives; their prompts are in Appendix D. External-baseline versions and evaluation protocols are detailed in Appendix N.2.

Table 2: **Evaluation on Peacekeeper Elite.** We compare success rate (SR, %) and inference latency (Lat., s). 3A-VLA uses zero-shot Qwen2-VL-7B IA, pretrained SegFormer-B2 ESA, and AAR. The controlled IL variants use Qwen2-VL-7B. Table 32 reports available standard deviations for SR. CombatVLA is our reproduction and Open-P2P is adapted (Appendix N.2).

Model	Parachuting	Scavenging	Combat	Revival	Vehicle	Rotation	Avg. ↑	Lat. ↓
Proprietary Foundation Models								
GPT-4o [22]	41.4	31.4	18.6	47.1	38.6	40.0	36.2	1.6
Claude-3.7-Sonnet [1]	48.6	38.6	24.3	57.1	41.4	45.7	42.6	2.2
Gemini-2.5-Pro [13]	55.7	42.8	25.7	58.6	47.1	51.4	46.9	1.5
Open-Source Vision-Language Models								
Qwen2-VL-7B [32]	22.9	15.7	8.6	27.1	18.6	21.4	19.1	1.1
Vision-Language-Action Models								
Vanilla IL	54.3	48.6	44.3	68.6	51.4	52.9	53.4	1.1
IL + CoT	55.3	49.1	45.7	68.1	53.1	53.9	54.2	1.1
IL + Image Caption	56.0	50.4	46.3	68.7	53.4	54.3	54.9	1.1
IL + CoT + Caption	56.3	50.9	46.8	69.3	53.7	55.0	55.3	1.1
CombatVLA [9]	48.1	44.7	42.1	44.0	47.1	49.0	45.8	–
Open-P2P [36]	48.7	45.2	63.5	54.6	58.3	60.1	55.1	–
3A-VLA (Ours)	**66.1**	**60.5**	**67.1**	**70.2**	**68.9**	**65.7**	**66.4**	**1.1**

4.2 Main results: Analysis of agent behaviors

Evaluation on Minecraft. Table 1 shows the results for the three Minecraft task groups. 3A-VLA achieves the best overall performance: it attains the highest OOD success rates across all three task groups while requiring the fewest steps. The improvement is most pronounced in "Combat", where 3A-VLA increases OpenHA's success rate from 40.0% / 31.9% to 45.9% / 35.4% on "Mini/All" splits—demonstrating the value of abstraction under rapid, cluttered observations. The gains extend beyond Combat to Embodied tasks, where the Mini/All SR increases from OpenHA's 37.0/30.1% to 40.6/33.5%, and GUI tasks, where it rises from 33.3/32.5% to 34.7/34.8%. In Combat, the average number of steps decreases from 316 to 271, indicating more efficient execution. The relatively smaller gains on GUI suggest that abstraction is particularly beneficial for visually dynamic control tasks.

Evaluation on Peacekeeper Elite. As shown in Table 2, 3A-VLA achieves state-of-the-art performance on the Peacekeeper Elite benchmark. The complete IA+ESA+AAR model reaches 66.4% versus 55.3% for IL+CoT+Caption, a gain of 11.1 percentage points, with particularly large gains in combat and vehicle acquisition. These comparisons support the benefit of the combined abstraction and reweighting pipeline. Compared with Vanilla IL, 3A-VLA improves the average SR by 13.0 percentage points (53.4% to 66.4%). Against the stronger IL+CoT+Caption baseline, the largest gains occur in Combat (46.8% to 67.1%, +20.3 points) and Vehicle acquisition (53.7% to 68.9%, +15.2 points), whereas Revival improves only modestly (69.3% to 70.2%, +0.9 points). This non-uniform improvement is consistent with greater benefits in visually dynamic, action-sensitive tasks.

4.3 Mechanism analysis

Individual and combined contributions. On Minecraft, IA alone improves SR from 24.6% to 30.2%, while ESA alone reaches 30.6%. Combining IA and ESA further raises SR to 36.5% without AAR, and the full model achieves 40.4%. These results demonstrate the complementary benefits of the two abstractions, with AAR providing an additional improvement beyond their joint supervision.

Analysis of representation sparsity in IA and ESA. Table 3 validates our "less is more" hypothesis by comparing sparse abstractions against their denser counterparts within an identical pipeline. **Intention Abstraction (IA):** Contrary to the intuition that more reasoning helps, keyword-based IA surpasses CoT variants. This suggests that concise keyword semantics effectively capture reusable intent primitives, circumventing the performance degradation associated with redundant text related to the rationale. **Environment Abstraction (ESA):** Our latent ESA outperforms dense alternatives—including action-conditioned image prediction ("Img. Pred.") and global semantic segmentation ("Full Sem.")—as well as non-pixel-level scene descriptions like image captions ("Img. Caption").

Table 3: **Ablation on IA/ESA** and alternatives. Sparse keyword IA and latent ESA constitute our proposed abstractions; the bottom row shows our full approach.

Variant	Alternatives		Minecraft	
	IA	ESA	Steps ↓	SR ↑
Baseline	–	–	332	24.6
	Full CoT	Img. Caption	309	32.3
IA	Full CoT	–	310	28.3
	Keywords	–	298	30.2
ESA	–	Img. Caption	320	27.0
	–	Img. Pred.	322	26.8
	–	Full Sem.	318	27.4
	–	**Latent**	304	30.6
Ours (w/o AAR)	**Keywords**	**Latent**	297	36.5
Ours	**Keywords**	**Latent**	**285**	**40.4**

Table 4: **Ablation on AAR** and alternatives. We compare *manual* and *learned* weighting schemes.

Scheme	Minecraft	
	Steps ↓	SR ↑
Ours (w/o AAR)	297	36.5
Fixed w=0.33	303	36.9
Fixed w=0.67	302	37.0
Scalar cosine	299	37.1
Scalar $\odot$	295	37.6
CoT+Cap. w/o AAR	309	32.3
CoT+Cap. w/ VLM-Score	307	32.8
CoT+Cap. w/ AAR	301	36.0
AAR w/o $\bar{H}$	291	38.2
AAR w/o $\mathcal{D}$	289	37.6
AAR (Ours)	**285**	**40.4**

Table 5: **Zero-shot cross-game combat transfer** (Hit SR, %). All models are trained on Peacekeeper Elite combat data via imitation learning and evaluated without target-game fine-tuning. 3A-VLA uses the simplest IA/ESA pipeline. Full Hit/Kill results, external baselines, and run-to-run uncertainty are provided in Appendix K.1.

Method	Valorant	Minecraft	CS2	GTA V	Elden Ring	Mount & Blade
Qwen2-VL-7B (Vanilla IL)	27.1	8.6	18.6	15.7	7.1	8.6
Qwen2-VL-7B (IL + CoT + Caption)	35.1	17.7	28.3	26.0	13.2	16.1
3A-VLA (Ours)	**42.9**	**25.7**	**40.7**	**38.6**	**21.4**	**24.3**

While captions provide spatial cues, they are often encumbered by redundant linguistic details. In contrast, ESA operates at the pixel level, filtering transient noise to prioritize structurally actionable information through coarse affordance grids, rather than pursuing dense reconstruction fidelity. With AAR off and backbone, trajectories, semantic scope, and a 100-token target budget matched, IA+ESA reaches $36.5/62.9\%$ MC/PE SR versus $32.4/55.0\%$ for Unified and higher means than two alternative splits. Appendix P provides the controls and prompts.

Adaptive loss reweighting via AAR. Table 4 ablates AAR within the IA-ESA pipeline. **(i)** The "Fixed" rows apply a discrete threshold: if the semantic keywords from IA and ESA are non-overlapping, we manually down-weight the action loss by a static factor (w=0.33 or 0.67) rather than a learned weight. Scalar cosine and $\odot$ replace this discrete rule with simple heuristic alignment scores (0∼1). These manual designs recover part of the gain, but all remain below learned AAR, showing that IA–ESA alignment should modulate action imitation in a continuous and adaptive manner.

(ii) AAR remains effective even with degraded IA/ESA surrogates like "CoT+Cap," consistently outperforming naive weighting strategies that rely on an additional VLM for direct action loss scoring ("VLM-Score"; prompt in Appendix E). **(iii)** Removing the entropy regularizer $\bar{H}$ or stop-gradient $\mathcal{D}$ degrades performance, as AAR works best when the weight is learned from the relation feature without collapsing onto a few samples or directly perturbing the backbone representation. The integration of the IA, ESA, and learned AAR achieves the strongest performance, underscoring that sparse abstraction and adaptive reweighting are mutually reinforcing. Appendix H provides a comprehensive statistical analysis of the learned AAR weights, offering insights into their interpretability across diverse scenarios, alongside an ablation study on hyperparameters and robustness validation. With IA+ESA fixed, AAR reaches $40.4 \pm 1.5\%$ MC SR versus $37.5 \pm 1.7\%$ for the strongest loss-only alternative (Appendix H.1). Its score is a learned relation-based weighting signal, not calibrated demonstration reliability.



Figure 3: **Attention and zero-shot generalization.** Trained on Peacekeeper Elite, the IA, ESA, and AAR progressively sharpen the focus on task-relevant regions as the attention entropy decreases, enabling the model to suppress irrelevant visual distractors even in unseen games (rows 2 and 3).

Emergent attention sparsity as a side effect. Figure 3 shows more concentrated attention with abstraction supervision: entropy decreases from $H = 7.58$ to $H = 3.42$ in the first example. Appendix I reports layerwise diagnostics. Section 4.6 evaluates whether this concentration supports token pruning, comparing connectivity scoring with generic pruning methods and reporting the accuracy–latency trade-off.

4.4 Zero-shot combat transfer and robustness

Table 5 retains Hit SR for PE-combat-trained models without target-game adaptation. 3A-VLA attains the highest mean Hit SR in all six games, e.g., 42.9% versus 39.4% for adapted Open-P2P in Valorant (Appendix K.1), with the largest relative gains over Vanilla IL in visually distant titles such as Minecraft (8.6% to 25.7%). Complete Hit/Kill evaluation reports 3A-VLA Kill SR of 10.6–21.4% across six games (Appendix K.1). In opposing-instruction controls, 3A-VLA reaches 7.5/14.3/17.8% Both-Correct on wall/doorway/enemy conflicts versus 2.3/4.2/5.1% for random instructions, supporting partial instruction-conditioned disambiguation with a remaining MC in-domain gap (Appendix K.2). The evidence supports combat transfer under this protocol; target-title exposure during pretraining cannot be excluded. Figure 3 gives illustrative attention maps; GameVerse/GameWorld provide complementary task-specific evidence in Appendices M.2 and M.3. Appendix L provides failure cases and diagnostics, highlighting persistent challenges in occlusion, long-horizon recovery, and domain-specific interactions.

4.5 Component cost-effectiveness analysis

Table 6 shows that the simplest IA+ESA+AAR pipeline achieves 40.4% SR with approximately 8 GPU-hours of annotation, improving over the baseline by 15.8 percentage points (about 5 GPU-hours for IA and 3 for ESA, with no API cost or manual labeling). It also exceeds either enhanced component alone (34.8% for IA; 32.5% for ESA). Enhancing both annotators further raises SR to 45.0% (+4.6 points).

Table 26 in Appendix M further shows gains with Qwen3-VL-8B on both Minecraft (+16.2 points) and Peacekeeper Elite (+10.5 points), supporting applicability across backbones and environments.

4.6 Pruning strategy comparison

Figure 4 illustrates the robustness of the enhanced IA+ESA configuration (without AAR). While JARVIS-VLA degrades substantially under increasing pruning ratios, our abstraction-trained representation retains most of its performance. At 75% pruning, connectivity-based selection achieves 41.5% SR compared with 42.1% without pruning, with approximately 4× faster inference.

Table 7 further compares connectivity scoring with FastV and ToMe under the same model configuration and latency budget. The baseline degrades under the tested pruning ratios. At equal pruning

Table 6: **Cost-effectiveness analysis on Minecraft.** Each row reports a single configuration; the cost column shows incremental annotation effort beyond the trajectory dataset.

Tier	Configuration	MC SR (%)	Cost
Baseline	No IA, no ESA, no AAR	24.6	—
Simplest	IA only (Qwen2-VL-7B, no RAG)	30.2	≈5 GPU-hrs
Simplest	ESA only (zero-shot SegFormer-B2)	30.6	≈3 GPU-hrs
Simplest	IA + ESA	36.5	≈8 GPU-hrs
Simplest (Ours)	**IA + ESA + AAR**	**40.4**	≈8 GPU-hrs
Enhanced	IA only (GPT-4o + full RAG)	34.8	≈$260
Enhanced	ESA only (SegFormer-B2 finetuned)	32.5	≈16 person-hrs
Enhanced	IA + ESA	42.1	≈$260 + 16 p-hrs
Enhanced	**IA + ESA + AAR**	**45.0**	≈$260 + 16 p-hrs

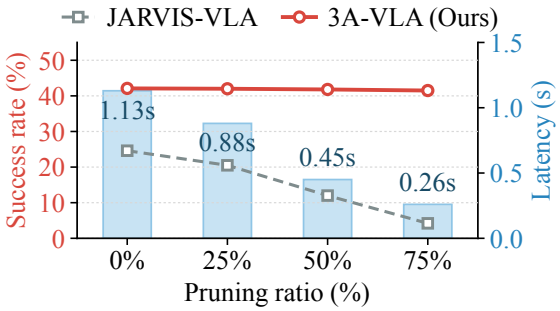


Figure 4: **Robustness to token pruning.** 3A-VLA maintains success rate at 75% pruning with a 4× inference speedup.

Table 7: **Pruning on Minecraft.** 75% pruning with enhanced IA+ESA; Connectivity scoring achieves the best success rate among pruning methods.

Method	SR (%)	Latency (s)
No pruning	42.1	1.1
Connectivity	**41.5**	**0.3**
FastV [8]	39.2	0.3
ToMe [5]	38.6	0.3
Magnitude	36.4	0.3
Random	30.9	0.3

ratios, FastV and ToMe reach 39.2% and 38.6% SR, compared with 41.5% for connectivity scoring. Our contribution is therefore not a new pruning algorithm, but an abstraction training recipe that makes the representation structurally prunable—the connectivity score simply exploits that structure.

5 Conclusion

We present 3A-VLA, an abstraction-aligned action learning framework that addresses perceptual overload in dynamic game environments via a cognitive bottleneck paradigm. By decoupling decision-making through intention (IA) and environment (ESA) abstractions, we shift VLA grounding from low-level correlations to deep semantic alignment. The AAR module further enables adaptive synergy between fine-grained control and high-level reasoning without manual rules. Experiments on Minecraft and Peacekeeper Elite demonstrate improved task success over the evaluated baselines, alongside zero-shot combat transfer to unseen games. Controlled ablations support the complementary roles of sparse abstraction and adaptive supervision, while comparisons with denser alternatives highlight the importance of abstraction content. Because IA and ESA serve only as auxiliary training targets, these gains preserve the standard VLA inference pipeline. Together, the results suggest that compact, action-relevant representations and alignment-aware supervision offer a promising design principle for transferable game agents.

Limitations and future work. 3A-VLA remains single-frame and reactive, lacking explicit temporal memory and long-horizon planning, which limits autonomy in extended FPS tasks. Moreover, ESA lacks explicit terrain and collision geometry, limiting fine-grained traversability reasoning. In future work, the integration of intrinsic motivation will be explored to extend 3A-VLA toward an autonomous strategic discovery model.

Acknowledgments and Disclosure of Funding

This work was supported by Tencent through the provision of computational resources, data infrastructure, and research support. Additional funding was provided by the National Natural Science Foundation of China (Grant No. 52375271) and the "Pioneer and Leading Goose" R&D Program of Zhejiang Province (Grant No. 2024C01044).

Competing interests. The authors declare no competing interests beyond the affiliations and support disclosed above.

References

[1] Anthropic. Claude 3.7 Sonnet and Claude Code, 2025.

[2] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In *NeurIPS*, 2022.

[3] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *CoRL*, 2025.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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. In *RSS*, 2025.

[5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *RSS*, 2023.

[7] Shaofei Cai, Zihao Wang, Kewei Lian, Zhancun Mu, Xiaojian Ma, Anji Liu, and Yitao Liang. ROCKET-1: Mastering open-world interaction with visual-temporal context prompting. In *CVPR*, 2025.

[8] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth $1/2$ tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 2024.

[9] Peng Chen, Pi Bu, Yingyao Wang, Xinyi Wang, Ziming Wang, Jie Guo, Yingxiu Zhao, Qi Zhu, Jun Song, Siran Yang, et al. CombatVLA: An efficient vision-language-action model for combat tasks in 3D action role-playing games. In *ICCV*, 2025.

[10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.

[11] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. MineDojo: Building open-ended embodied agents with internet-scale knowledge. In *NeurIPS*, 2022.

[12] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. In *RSS*, 2024.

[13] Google. Gemini 2.5: Our most intelligent AI model, 2025.

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[15] Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom M. Mitchell, and Eric P. Xing. Learning data manipulation for augmentation and weighting. In *NeurIPS*, 2019.

[16] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Pannag Sanketi, Quan Vuong, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, 2025.

[17] Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *ICML*, 2022.

[18] Muyao Li, Zihao Wang, Kaichen He, Xiaojian Ma, and Yitao Liang. JARVIS-VLA: Post-training large-scale vision language models to play visual games with keyboards and mouse. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

[19] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022.

[20] Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. STEVE-1: A generative model for text-to-behavior in Minecraft. In *NeurIPS*, 2023.

[21] Jiahao Lin, Weixuan Peng, Bojia Zi, Yifeng Gao, Xianbiao Qi, Xingjun Ma, and Yu-Gang Jiang. BrokenVideos: A benchmark dataset for fine-grained artifact localization in AI-generated videos. In *ACM MM*, 2025.

[22] OpenAI. Hello GPT-4o, 2024.

[23] Mingyu Ouyang, Siyuan Hu, Kevin Qinghong Lin, Hwee Tou Ng, and Mike Zheng Shou. GameWorld: Towards standardized and verifiable evaluation of multimodal game agents. In *ECCV*, 2026.

[24] Konstantinos P. Panousis, Dino Ienco, and Diego Marcos. Coarse-to-fine concept bottleneck models. In *NeurIPS*, 2024.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[26] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[27] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-Weight-Net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019.

[28] Weihao Tan, Xiangyang Li, Yunhao Fang, Heyuan Yao, Shi Yan, Hao Luo, Tenglong Ao, Huihui Li, Hongbin Ren, Bairen Yi, Yujia Qin, Bo An, Libin Liu, and Guang Shi. Lumine: An open recipe for building generalist agents in 3D open worlds. *arXiv preprint*, 2025.

[29] Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. Cradle: Empowering foundation agents towards general computer control. In *ICML*, 2025.

[30] Rufin VanRullen and Ryota Kanai. Deep learning and the global workspace theory. *Trends in Neurosciences*, 2021.

[31] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

[32] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint*, 2024.

[33] Zihao Wang, Xujing Li, Yining Ye, Junjie Fang, Haoming Wang, Longxiang Liu, Shihao Liang, Junting Lu, Zhiyong Wu, Jiazhan Feng, Wanjun Zhong, Zili Li, Yu Wang, Yu Miao, Bo Zhou, Yuanfan Li, Hao Wang, Zhongkai Zhao, Faming Wu, Zhengxuan Jiang, Weihao Tan, Heyuan Yao, Shi Yan, Xiangyang Li, Yitao Liang, Yujia Qin, and Guang Shi. Game-TARS: Pretrained foundation models for scalable generalist multimodal game agents. *arXiv preprint*, 2025.

[34] Zihao Wang, Muyao Li, Kaichen He, Xiangyu Wang, Zhancun Mu, Minghao Liu, Anji Liu, and Yitao Liang. OpenHA: A series of open-source hierarchical agentic models in Minecraft. In *ICML*, 2026.

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.

[36] Yuguang Yue, Irakli Salia, Samuel Hunt, Chris Green, Wenzhe Shi, and Jonathan J. Hunt. Scaling behavior cloning improves causal reasoning: An open model for real-time video game playing. *arXiv preprint*, 2026.

[37] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Proceedings of The 8th Conference on Robot Learning*, 2025.

[38] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[39] Kuan Zhang, Dongchen Liu, Qiyue Zhao, Jinkun Hou, Xinran Zhang, Qinlei Xie, Miao Liu, and Yiming Li. GameVerse: Can vision-language models learn from video-based reflection? In *ICML*, 2026.

[40] Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Qiang Fu, Haobo Fu, Zilong Zheng, and Yitao Liang. MCU: An evaluation framework for open-ended game agents. In *ICML*, 2025.

[41] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the Minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint*, 2023.

[42] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

A Minecraft data curation

To ensure our model generalizes across diverse 3D embodied environments, we incorporate data from *Minecraft*, a domain offering high structural freedom and long-horizon planning challenges. We leverage the large-scale public dataset released by the Video PreTraining (VPT) project [2]. Specifically, we follow JARVIS-VLA [18] and OpenHA [34] for our frame-action pair, which consists of high-quality human gameplay trajectories. Unlike random web-scraped video, this subset includes accurate ground-truth actions labeled via an Inverse Dynamics Model (IDM), ensuring the data represents meaningful, goal-oriented behaviors (e.g., resource gathering, crafting, and combat).

B Peacekeeper Elite data curation

We introduce a dataset based on *Peacekeeper Elite*, a complex, highly dynamic FPS game. The 200-hour PE corpus was collected by internal full-time data-collection and annotation staff in operator-authorized dedicated test sessions, with informed consent under the stated IRB-approved protocol. It was not mined from public matchmaking or ordinary players' existing matches. All human participants in these sessions were consenting staff; no uninformed opponents or non-consenting minors were involved. Appendix O documents the task brief, compensation, privacy, and release boundaries.

B.1 Replay data pipeline

We build a scalable data curation pipeline on top of in-game replay logs. Most modern FPS titles are developed with Unreal Engine (UE), which provides strong native support for replay recording and playback. A replay is a structured record generated during gameplay that captures match-wide trajectories and events for all players, and can be rendered by a playback client to reproduce the corresponding video.

From replay logs, we parse per-timestep player states—including world position, view orientation, and status flags—as well as discrete in-match events. When an event satisfies predefined criteria or a task is completed, we retrieve a short temporal context window by backtracking 10–30 seconds and extract a 10–30 second clip. Each clip is then processed to produce (1) a rendered video segment and (2) aligned action signals for model training. We apply this pipeline only to the controlled sessions described above. Replay-wide fields cover participating research staff, not uninformed public players. Structured and visible identifiers, including kill-feed names, are masked before training; replay-derived action extraction requires no additional manual frame-by-frame action annotation.

B.2 Action extraction

Within each mined clip, we discretize the timeline into fixed intervals of $\Delta t = 0.25$s (4 Hz). For each interval, we compute changes in camera orientation and displacement, and extract whether key actions occur within the interval.

Camera control. Let $(\text{yaw}_t, \text{pitch}_t)$ denote the view orientation at the sampled boundary. We define the camera deltas as:

$$\Delta\text{yaw}_t = \text{yaw}_{t+\Delta t} - \text{yaw}_t, \qquad \Delta\text{pitch}_t = \text{pitch}_{t+\Delta t} - \text{pitch}_t, \tag{6}$$

and quantize them into discrete bins based on the screen aspect ratio and the empirical distribution of player inputs:

$$a_t^{\text{yaw}} = \text{Bin}_{\text{yaw}}(\Delta\text{yaw}_t), \qquad a_t^{\text{pitch}} = \text{Bin}_{\text{pitch}}(\Delta\text{pitch}_t). \tag{7}$$

Movement control. For movement, we discretize the heading direction into eight compass directions plus stationary, yielding 9 categories:

$$a_t^{\text{move}} \in \{0, 1, \ldots, 8\}, \tag{8}$$

corresponding to $\{0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°\}$ and no movement.

Button actions. We further encode action buttons as binary indicators. For *sustained actions* (e.g., Aim/Crouch/Prone), we mark a button press only when the corresponding state changes between adjacent intervals. For *instantaneous actions* (e.g., Jump/Fire), we mark a press if the event occurs at any time within the interval. All button actions are represented as Boolean variables.

Overall, each 0.25s interval is represented by a discrete action tuple:

$$\mathbf{a}_t = \big(a_t^{\text{yaw}},\ a_t^{\text{pitch}},\ a_t^{\text{move}},\ \mathbf{b}_t\big), \tag{9}$$

where $\mathbf{b}_t \in \{0,1\}^K$ denotes K binary button actions (e.g., aim, fire, mark, jump, crouch, prone, grenade/throw, enter/exit vehicle, heal, etc.).

C Text-based action space

To bridge the modality gap, we map the discrete action tuples $\mathbf{a}_t$ (derived in Appendix B) into a **plain text action space**. Following Lumine [28] and OpenHA [34], we linearize the signals into a plain text calling format:

1. View control: `move(`d_x`, `d_y`)` The quantized bins a_t^{yaw} and a_t^{pitch} are mapped to integer arguments representing the horizontal and vertical screen-space pixel displacements (d_x, d_y) of the mouse for the `move` function.

2. Mouse interaction: `click(button)` Mapped from $\mathbf{b}_t$ elements:

- `click('left')`: Attack/Fire.
- `click('right')`: Aim Down Sights (ADS).

3. Keyboard interaction: `press(key)` Mapped from a_t^{move} and remaining $\mathbf{b}_t$ elements:

- **Movement:** a_t^{move} is converted to combinations of `'w', 's', 'a', 'd'`.
- **Actions:** `'space'` (jump), `'c'` (crouch), `'f'` (interact), etc.

Multi-action syntax. Multiple actions occurring in the same timestep are connected with commas (e.g., `Action: press('w'), click('left')`), creating a unified text corpus for cross-domain training.

D Chain-of-thought (CoT) and image caption prompt

CoT and image-caption annotations are offline hindsight supervision signals generated from the instruction and a 4s trajectory segment. The generated text is appended after the action chunk during training and fully truncated at inference.

Default CoT prompt.

```
System: You are an reasoning annotator for embodied game
trajectories. Given first-person visual frames and the task
instruction, explain the agent's current situation and likely action
rationale in a concise chain-of-thought paragraph. Use observable
evidence and avoid unsupported assumptions.
Input: [Instruction], [Sampled frames from 4s trajectory]
Output: Chain-of-thought reasoning.
```

This is the prompt used for the **main-text "CoT"** baseline: no external knowledge base is queried, and no retrieved context is provided to the annotator.

Default image-caption prompt.

```
System: You are an image captioner for embodied game trajectories.
Given first-person visual frames and the task instruction, summarize
```

```
the visible scene including key object locations, spatial layout,
and immediate context in one concise paragraph.  Describe observable
visual content rather than intentions or reasoning.
Input:  [Instruction], [Sampled frame]
Output:  Concise image-caption paragraph only.
```

This is the prompt used for the **main-text "Image-caption"** baseline: no external knowledge base is queried, and no retrieved context is provided to the annotator.

E Loss reweight score prompt

For the **CoT+Cap. w/ VLM-Score** baseline in Table 4 of the main-text, we use an offline VLM to directly assign a sample-wise action-loss reweighting score. This prompt is constructed by concatenating the default CoT prompt and the default image-caption prompt above, then appending an action-importance scoring request for the current ground-truth action.

CoT+caption action-importance prompt.

```
System:  You are an reasoning annotator for embodied game
trajectories.  Given first-person visual frames and the task
instruction, explain the agent's current situation and likely action
rationale in a concise chain-of-thought paragraph.  Use observable
evidence and avoid unsupported assumptions.  Then summarize the
visible scene including key object locations, spatial layout, and
immediate context in one concise image-caption paragraph.  Describe
observable visual content rather than intentions or reasoning in the
caption.  Finally, given the current action, rate how important
this action should be for training the action imitation loss
on a continuous score from 0 to 1, where 0 means the action is
weakly related or ambiguous under the current context and 1 means
the action is essential and well supported by the CoT and visual
caption.
Input:  [Instruction], [Sampled frames from 4s trajectory], [Current
action]
Output:  Chain-of-thought reasoning.  Image-caption paragraph.  Loss
reweight score:  <a single number in [0,1]>.
```

The resulting scalar is normalized within each mini-batch and applied only to the action imitation term, giving a direct VLM-scored alternative to learned AAR.

F IA annotation pipeline and noise analysis

Our main-text results use a *zero-shot Qwen2-VL-7B* IA annotator with no retrieval augmentation. Here we expose the full pipeline used for the enhanced IA variant, and quantify sensitivity to annotator choice and retrieval quality.

F.1 Annotation prompts

IA is an *offline hindsight supervision signal* generated from the instruction and a 4s trajectory segment. The resulting keyword sequence $\mathbf{y}_{\text{ia}}$ is appended after the action chunk during training, and is fully truncated at inference (no online leakage). The main-text default uses one zero-shot Qwen2-VL-7B pass with no retrieval augmentation; if the annotator internally produces explanatory text, we discard it and keep only the final keyword sequence.

Default IA prompt.

```
System:  You are an reasoning annotator for embodied game
trajectories.  Given first-person visual frames and the task
instruction, explain the agent's current situation and likely action
rationale in a concise chain-of-thought paragraph.  Use observable
evidence and avoid unsupported assumptions.  Output only 3-5 short
```

```
keywords.  Use reusable task primitives rather than full sentences
or chain-of-thought.
Input:  [Instruction], [Sampled frames from 4s trajectory]
Output:  Comma-separated keywords only.
```

This is the prompt used for the **main-text IA** results: no external knowledge base is queried, and no retrieved context is provided to the annotator. Notably, this IA prompt differs from the CoT prompt (D) solely in its output specification: it requires concise, predefined task primitives instead of free-form reasoning chains.

Keyword vocabulary. After normalizing and deduplicating free-form LLM/VLM outputs, we obtain ≈120 unique keywords in Minecraft and ≈80 in Peacekeeper Elite, organized into the categories below.

Table 8: IA keywords (Minecraft).

Category	Keywords
Combat	enemy front, low health, attack zombie, creeper near
Resource	mine wood, dig dirt, harvest stone, search iron
Crafting	craft pickaxe, open inventory, place table, make sword
Navigation	jump obstacle, swim water, run forward, enter cave

Table 9: IA keywords (Peacekeeper Elite).

Category	Keywords
Combat	shoot enemy, target locked, reload weapon, take cover
Movement	sprint forward, jump wall, crouch, navigate building
Vehicle	drive car, steer left, brake, enter vehicle
Status	heal, use bandage, low ammo, zone closing

F.2 Enhanced retrieval-augmented generation (RAG) variant.

For the enhanced IA setting only, we test the following retrieval-augmented two-stage pipeline. This variant is used for sensitivity and upper-bound studies, not for the main-text results.

Prompt 1 (intention reasoning).

```
System:  Based on visual frames and instruction, infer the agent's
core intention in a concise paragraph focusing on the high-level
goal.
Input:  [Instruction], [Sampled frames from 4s trajectory]
Output:  Short paragraph describing the agent's likely intention and
supporting evidence.
```

The reasoning output is used as a query for RAG vector retrieval, returning the most relevant wiki entries as context for keyword extraction.

Prompt 2 (intention abstraction).

```
System:  Based on visual frames, instruction, and retrieved wiki
knowledge, extract 3-5 concise keywords representing current intent
and environmental state.
Input:  [Instruction], [Wiki Knowledge]
Output:  Comma-separated keywords only.
```

RAG knowledge base. For the enhanced IA variant, the RAG database contains ≈270K entries for Minecraft and ≈200K for Peacekeeper Elite, sourced from official wikis.

Annotator consistency. Three independent GPT-4o runs on 500 random segments achieve ≈87% keyword Jaccard similarity, confirming that extraction is stable across re-queries.

F.3 IA noise / cost sensitivity

To probe IA robustness, we ablate annotator model, retrieval quality, and spend. All rows use ESA off to isolate the IA channel.

Table 10: **IA annotation analysis (ESA off).** The pipeline degrades gracefully with noisier retrieval and a weaker backbone. The zero-shot Qwen2-VL setting used in the main text (bold row) already improves substantially over baseline. Mean±std is shown for the no-retrieval teacher controls where reported; other entries remain point estimates.

Setting	Model	RAG correct ratio	MC SR (%)	Cost
Baseline	—	—	24.6±1.8	0
IA	GPT-4o	0% (w/o RAG)	31.8±1.6	≈$150
IA	GPT-4o	40% (noisy RAG)	32.4	≈$260
IA	GPT-4o	70% (noisy RAG)	33.1	≈$260
IA	GPT-4o	100% (full RAG)	34.8	≈$260
IA (main text)	**Qwen2-VL-7B**	**0% (w/o RAG)**	**30.2±1.8**	**≈5 GPU-hrs**
IA	Qwen2-VL-7B	100% (full RAG)	33.8	≈8 GPU-hrs

The monotonic trend (30.2 → 33.8 → 34.8) confirms that richer retrieval and stronger backbones provide additive gains, but even the cheapest fully-local setup reaches 30.2%, well above the 24.6% baseline.

Both no-retrieval annotators improve over no IA (Table 10), supporting a benefit beyond dependence on a single teacher family. These teacher-only controls keep ESA disabled; they are not the full model.

G ESA segmentation noise analysis

Our main-text ESA uses zero-shot SegFormer-B2. To test how segmentation quality affects the downstream policy, we inject synthetic structural noise—random boundary erosion/dilation and semantic class swapping—and compare against a variant finetuned on 1,000 domain frames (test mIoU measured on 200 held-out frames). Throughout we keep IA off to isolate the ESA channel.

Table 11: **ESA annotation analysis (IA off).** A 32.5-point mIoU gap (from 58.9 heavy-noise to 91.4 finetuned) maps to only a ∼4-point SR gap, because the 8×8 rank-pooled grid smooths pixel-level errors while preserving foreground structure.

Setting	mIoU (%)	MC SR (%)	Cost
Baseline (no IA, no ESA)	—	24.6	0
SegFormer-B2 + heavy structural noise	58.9	28.4	≈3 GPU-hrs
SegFormer-B2 + mild structural noise	70.2	29.7	≈3 GPU-hrs
SegFormer-B2 w/o finetune (main text)	**83.7**	**30.6**	**≈3 GPU-hrs**
SegFormer-B2 (enhanced, finetuned)	91.4	32.5	≈16 person-hrs

The PE→Valorant evaluation (Table 5) uses no target-game segmentation data. This supports transfer under the reported combat protocol, rather than proving fully domain-invariant affordances.

H Analysis of abstraction alignment reweighting (AAR)

AAR score distribution and action-supervision reliability. We further analyze the task-dependent behavior of AAR by visualizing how it distributes action-imitation supervision over IA–ESA semantic

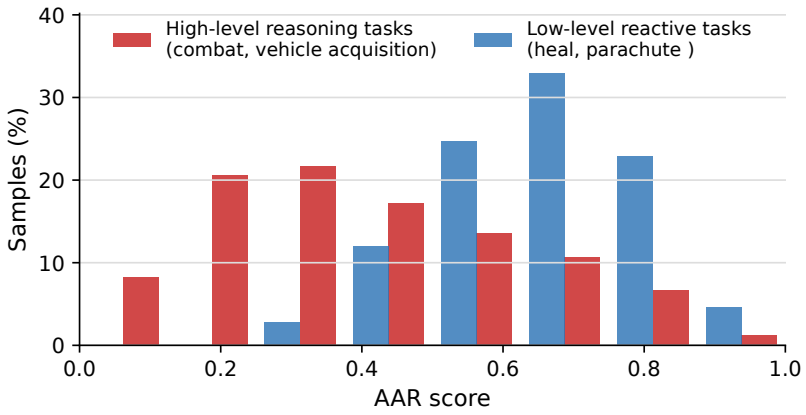


Figure 5: Distribution of AAR scores over IA–ESA semantic alignment. The horizontal axis is the AAR score $q_i = g_\psi(\mathbf{u}_i^{\mathrm{ia}} \odot \mathbf{u}_i^{\mathrm{esa}})$, where larger values yield larger batch-relative action weights. The semantic-alignment interpretation is correlational; the score is not a calibrated demonstration-reliability target. The vertical axis reports the percentage of samples in each score range. Low-level reactive tasks exhibit higher IA–ESA alignment, indicating that their action labels are more directly grounded in the current observation and should receive larger action-imitation weights. In contrast, high-level reasoning tasks show lower IA–ESA alignment, suggesting stronger observation-action ambiguity and therefore smaller low-level action-supervision weights.

alignment. For each demonstration sample, we compute the same relation score used by AAR:

$$\mathbf{r}_i = \mathbf{u}_i^{\mathrm{ia}} \odot \mathbf{u}_i^{\mathrm{esa}}, \qquad q_i = g_\psi(\mathbf{r}_i).$$

Here, $\mathbf{u}_i^{\mathrm{ia}}$ and $\mathbf{u}_i^{\mathrm{esa}}$ are the normalized span representations of IA and ESA, respectively, and $\odot$ denotes the Hadamard product. We group samples according to their AAR score q_i, as shown on the horizontal axis of Figure 5. Larger scores yield larger within-batch action-imitation weights. Their association with task type is consistent with semantic alignment, without independently measuring action-label reliability.

The vertical axis reports the percentage of samples falling into each AAR-score range. Thus, the figure visualizes how different task types are distributed over IA–ESA alignment levels. Although the AAR score is later used to compute action-imitation weights, Figure 5 focuses on the sample-level score distribution rather than the normalized supervision mass. In AAR, these scores are converted into batch-relative action-imitation weights:

$$p_i = \frac{\exp(q_i)}{\sum_{j=1}^{B} \exp(q_j)}, \qquad w_i = Bp_i.$$

Therefore, samples with higher IA–ESA alignment receive larger low-level action-supervision weights, while samples with weaker alignment are down-weighted.

Figure 5 shows a clear separation between low-level reactive tasks and high-level reasoning tasks. Low-level tasks, such as healing and parachuting, are concentrated in higher AAR-score regions. This pattern is consistent with more directly grounded short-horizon actions, but does not independently establish action-label reliability. As a result, AAR assigns these samples larger action-imitation weights. In contrast, high-level reasoning tasks, such as combat and vehicle acquisition, are more concentrated in lower AAR-score regions. For these tasks, the current observation is often insufficient to uniquely determine the demonstrated action. The same visual state may support multiple plausible high-level decisions, and the same high-level intention may be realized through different low-level action sequences. This motivates the interpretation that lower relation scores may be associated with observation-action ambiguity; the learned branch reduces their relative action-imitation weight. Multiple visual states may correspond to the same action, while a single visual state may admit multiple plausible actions. Therefore, low-alignment demonstrations can provide less reliable action supervision. AAR mitigates this issue by assigning larger action-imitation weights to samples whose intended objective and environment-salient affordances are semantically consistent.

This analysis supports the motivation of AAR. Rather than uniformly trusting all action labels, AAR adaptively reweights the action imitation term according to the learned IA–ESA relation. This is especially useful for long-horizon open-world tasks, where the mapping from visual observations to actions is inherently ambiguous.

Robustness to entropy regularization. To examine how sensitive AAR is to its entropy term, we sweep the regularization coefficient $\lambda_{\bar{H}}$ on Minecraft and report success rate in Figure 6. The resulting curve is notably flat: disabling the term entirely ($\lambda_{\bar{H}} = 0$) still reaches 38.2% SR, moderate regularization consistently helps, and the default setting $\lambda_{\bar{H}} = 10^{-2}$ attains the best 40.4% SR. Pushing the coefficient one order of magnitude higher ($\lambda_{\bar{H}} = 10^{-1}$) only drops SR to 39.0%, indicating that excessive smoothing of the batch-relative weights begins to wash out the contrast between highly and weakly aligned demonstrations. Taken together, the sweep shows that AAR does *not* require careful tuning of $\lambda_{\bar{H}}$: the default value provides a robust operating point that prevents weight collapse while preserving adaptive sample reweighting.

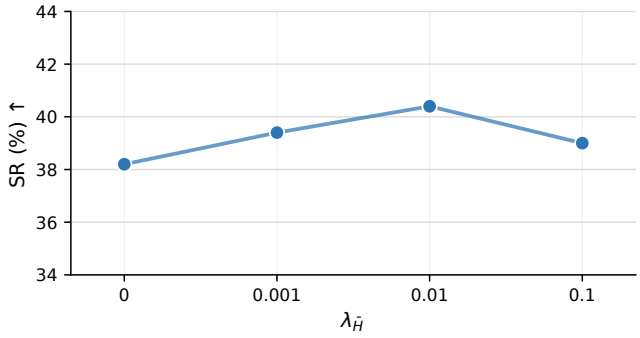


Figure 6: **Robustness to the entropy-regularization coefficient $\lambda_{\bar{H}}$.** Minecraft success rate as $\lambda_{\bar{H}}$ is swept over $\{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ in the AAR objective. Performance changes smoothly over the tested coefficient range and peaks at the default $\lambda_{\bar{H}} = 10^{-2}$, confirming that AAR benefits from mild entropy regularization but is otherwise insensitive to this hyperparameter.

Fixed action-loss weighting vs. AAR gating. We next ask whether the gains of AAR can be recovered by simply tuning a single global action-loss weight. Figure 7 contrasts two schemes across $w \in \{0.15, 0.33, 0.45, 0.67, 0.85, 1.0\}$: (i) a *no-strategy* baseline that uniformly scales all action losses by the same constant w; and (ii) a *fixed IA–ESA alignment-gated* variant that manually down-weights the action loss by w when the IA and ESA keywords are non-overlapping, while keeping aligned samples unweighted. These two hand-designed fixed-weight schemes are compared against the learned AAR in the main text, where AAR replaces the manually specified scalar w with a continuous sample-wise weight predicted from the IA–ESA relation feature. Both curves are unimodal and peak near w=0.5, but the gated curve sits uniformly above the no-strategy baseline over the entire sweep, and its two anchor points (w=0.33 and w=0.67) reproduce exactly the "Fixed" rows in Table 4 of the main text (36.9% and 37.0%). Crucially, even the best fixed weight remains several points below AAR (40.4% SR), which exploits continuous IA–ESA alignment to assign *per-sample* action-supervision weights rather than a single scalar. This confirms that a global weight can only deliver a coarse trade-off between action imitation and abstraction learning, whereas the adaptive reweighting in AAR is what drives the remaining gap.

Effect of fixed action-loss weighting. To further verify that the benefit of AAR does not simply come from globally rescaling the action imitation loss, we compare it with a fixed-weight variant. In this variant, we replace the adaptive sample-wise AAR weights w_i with a constant action-loss coefficient w for all samples:

$$\mathcal{L}_{\text{fix}} = \frac{1}{B} \sum_{i=1}^{B} \left[w \mathcal{L}_{\text{act}}^{(i)} + \mathcal{L}_{\text{abs}}^{(i)} \right].$$

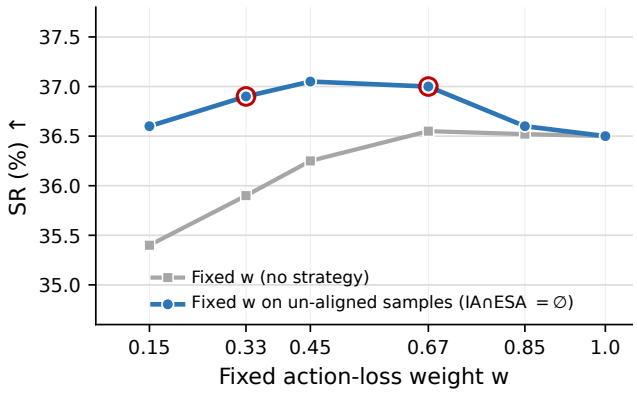


Figure 7: **Fixed action-loss weighting cannot replace adaptive AAR.** We sweep a constant action-loss coefficient $w \in \{0.15, 0.33, 0.45, 0.67, 0.85, 1.0\}$ under two schemes: a no-strategy baseline that applies w to all action loss (grey), and an AAR-alignment-gated variant that applies w only to the action loss when the IA and ESA keywords are non-overlapping (blue). The gated curve dominates the baseline throughout the sweep and matches the "Fixed" rows of Table 4 of the main text at w=0.33 and w=0.67, yet both schemes peak well below the AAR's 40.4% SR, showing that a single global weight is insufficient and that sample-adaptive reweighting is the key ingredient of the AAR.

We sweep w and report the Minecraft success rate in Figure 7. The performance is relatively stable when w lies in a moderate range, with the best result around $w = 0.5$. However, both overly small and overly large fixed action weights reduce performance. When w is too small, the model receives insufficient low-level action supervision; when w is too large, the action imitation term can dominate the abstraction losses and encourage overfitting to ambiguous action labels. This trend indicates that manually tuning a global action-loss weight provides only a coarse trade-off between action imitation and abstraction learning.

In contrast, AAR performs sample-wise reweighting according to the learned IA–ESA relation. Samples with strong semantic alignment between the intended abstraction and the environment-salient abstraction receive larger action-imitation weights, while weakly aligned and potentially ambiguous samples are down-weighted. Therefore, the advantage of AAR comes from adaptive semantic alignment-based weighting rather than from a fixed global rescaling of the action loss.

H.1 Comparison with loss-only reweighting

We keep IA+ESA fixed and replace only the action-loss weighting scheme. Loss-softmax uses $w_i \propto \exp(-\ell_i/\tau)$; the other controls use focal loss or inverse-focal/self-paced weighting.

Table 12: Action-loss weighting with IA+ESA fixed. AAR achieves higher mean success than the tested loss-only alternatives.

Weighting	MC SR (%)
No AAR	36.5 ± 1.7
Loss-softmax	37.3 ± 1.8
Focal loss	36.8 ± 1.9
Inverse-focal / self-paced	37.5 ± 1.7
AAR (Ours)	40.4 ± 1.5

AAR reaches $40.4 \pm 1.5\%$ MC SR versus $37.5 \pm 1.7\%$ for the strongest tested loss-only alternative, a 2.9-point difference in means. Together with Table 4, these results show that loss magnitude alone does not recover the full gain; they do not calibrate demonstration reliability. Learned weighting itself is not our novelty claim: L2RW [26] uses clean-set meta-gradients, Meta-Weight-Net [27] learns a loss-to-weight mapping from meta-data, Hu et al. [15] learn data manipulation, and Co-teaching [14]

exchanges selected examples between peers. AAR instead conditions batch-relative weights on detached IA–ESA relation features without a clean meta-set or peer model.

I Attention concentration and abstraction diagnostics

I.1 Attention concentration

To substantiate the "emergent concentration" claim, we measure Shannon entropy of the per-layer attention distribution over $N_{\text{vis}} = 336$ visual tokens. All variants are trained on identical data; entropy is mean±std over 3,000 held-out frames.

Table 13: **Attention entropy and success rate.** Lower entropy indicates more concentrated attention. The baseline is near the theoretical maximum $\log_2 N_{\text{vis}} \approx 8.39$; adding sparse IA/ESA supervision lowers entropy, and AAR further improves task performance under the main-text setting.

Variant	MC Entropy ↓	MC SR ↑	PE Entropy ↓	PE SR ↑
Baseline (Qwen2-VL IL)	7.76±0.10	24.6	7.71±0.04	53.4
+Image Caption auxiliary	7.28±0.12	27.0	7.33±0.08	54.9
+Simplest IA	6.14±0.31	30.2	6.37±0.24	57.8
+Simplest ESA	6.08±0.27	30.6	5.52±0.29	61.7
+Simplest IA+ESA+AAR (main text)	3.42±0.42	40.4	4.67±0.31	66.4
+Enhanced IA	5.24±0.39	34.8	5.53±0.32	61.4
+Enhanced ESA	5.89±0.22	32.5	5.11±0.38	64.3
Enhanced IA+ESA+AAR	**4.80±0.29**	**45.0**	**4.41±0.57**	**68.6**

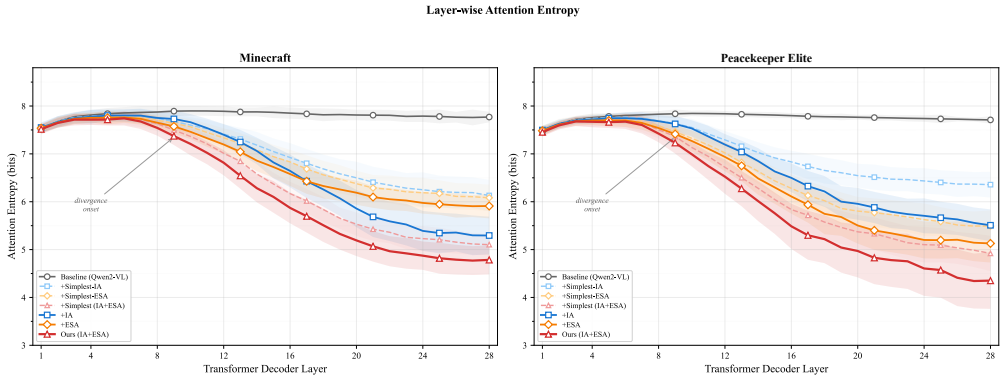


Figure 8: **Layer-wise attention entropy across all 28 transformer layers of Qwen2-VL-7B.** Entropy is averaged over 3,000 held-out Minecraft frames. The vanilla baseline (black) stays close to the theoretical maximum throughout, whereas sparse IA/ESA supervision drives down entropy especially in the mid-to-late layers where task-relevant grounding happens. The enhanced IA+ESA variant (red) concentrates attention sharpest in layers 12–22, matching the pattern that makes the representation structurally prunable (see Section 4.6).

At equal pruning ratios, FastV / ToMe keep entropy near baseline (∼7.7) with lower SR (39.2 / 38.6), confirming that concentration stems from IA/ESA/AAR training itself, not from the pruning step.

I.2 Why open-world inputs benefit from sparse IA/ESA

The main method uses IA and ESA as compact auxiliary targets, not as a separate bottleneck objective. Their purpose is to make the shared causal Transformer preserve the small set of intent and scene-affordance cues that disambiguate actions in open-world environments. This is especially important because visually diverse states often share the same short-horizon action but require different semantics:

- *Goal ambiguity*: "Enter the car" with no car in sight—the agent may walk forward regardless, but the intent determines what it should search for.
- *Strategic divergence*: near-identical PvP frames may require "fire", "take cover", or "retreat" depending on the instruction and semantic context.

We quantify intra-action visual diversity $d_{\cos}(\mathbf{X} \mid \mathbf{A})$ by computing frozen CLIP ViT-B/32 features per action bin and averaging pairwise cosine distances, weighted by bin frequency. Open-world domains have substantially larger diversity than tabletop manipulation (Tables 14 and 15), explaining why sparse intent keywords and coarse scene affordances provide useful auxiliary supervision.

Table 14: **Intra-action diversity across domains.** Open-world domains have 2–3× the intra-action diversity of tabletop settings, indicating that action labels alone are less informative about the policy-relevant state.

Dataset	Domain	$d_{\cos}(\mathbf{X} \mid \mathbf{A}) \uparrow$
LIBERO	Tabletop manipulation	0.062
Minecraft	Open-world sandbox	0.152
Peacekeeper Elite	Open-world FPS	0.188

Even restricted to semantically equivalent atomic actions ("no movement", "interaction"), LIBERO vs. MC/PE differs by $\sim 1.8\times$–$2.2\times$:

Table 15: **Per-action diversity.** Identical action semantics yield nearly double the visual diversity in open worlds vs. tabletop, motivating auxiliary intent and environment abstractions.

Action	LIBERO	MC	PE
No movement	0.098	0.178 (1.8×)	0.184 (1.9×)
Interaction	0.098	0.167 (1.7×)	0.215 (2.2×)

I.3 Abstraction content and AAR diagnostics

As described in Section 3.4, AAR learns a batch-relative action weight from the alignment between IA and ESA representations and applies it only to the action imitation term. The diagnostic table 16 below follows this current method: sparse IA and latent ESA help individually, while their combination with AAR gives the strongest main-text result.

Table 16: **Abstraction-content diagnostic under the current 3A-VLA objective.** Sparse keyword IA and latent ESA are complementary; AAR further improves the combined model by adaptively reweighting action imitation according to IA–ESA alignment. Numbers overlapping with the main tables follow the main-text values.

#	Method	MC SR ↑	PE SR ↑
1	Baseline (Qwen2-VL IL)	24.6±1.8	53.4±1.4
2	+Full CoT / rationale auxiliary	28.3±2.0	54.2±1.7
3	+Image-caption auxiliary	27.0±1.9	54.9±1.6
4	+Sparse keyword IA	30.2±1.8	57.8±1.5
5	+Latent ESA	30.6±1.7	61.7±1.4
6	**3A-VLA (IA+ESA+AAR)**	**40.4±1.5**	**66.4±1.2**

J ESA latent resolution analysis

In this section, we provide a detailed ablation study on the resolution of the Environment Semantic Abstraction (ESA) latent representation. We investigate how the granularity of the spatial grid impacts the agent's decision-making efficiency and success rate in the **Minecraft** environment.

J.1 Experimental setup

The input visual observation is a raw RGB image with a resolution of 640×360. We compare our default ESA setup (using an 8×8 latent grid) against:

- **Baselines:** A vanilla baseline without abstraction, a standard Full Image prediction, and a Full Semantic Map.
- **Dense ESA:** High-resolution latent grids (256×256 and 64×64).
- **Coarse ESA:** An extremely low-resolution grid (2×2).

J.2 Results and analysis

The results are summarized in Table 17. We report the average **success rate (%)** and **average steps** (trajectory length) on the Minecraft holdout evaluation set. Lower average steps indicate higher navigational efficiency.

Table 17: **Ablation on ESA latent resolution in Minecraft (enhanced pipeline).** We compare different spatial resolutions against baselines. Our 8×8 latent grid achieves the best trade-off, achieving higher mean SR than the baseline and the tested dense representations.

Method	Resolution / type	Grid size	Avg. steps ↓	Success rate (%) ↑
Baseline	–	–	332	24.6
Full image prediction	Pixel reconstruction	640×360	314	28.4
Full semantic map	Global semantic	640×360	312	29.1
ESA (dense)	Latent grid	256×256	313	28.6
ESA (mid)	Latent grid	64×64	305	30.8
ESA (enhanced)	**Latent grid**	$\mathbf{8 \times 8}$	**293**	**32.5**
ESA (coarse)	Latent grid	2×2	329	25.0

Latent vs. full image/semantic. As shown in Table 17, our sparse latent representation (8×8) outperforms dense alternatives in both efficiency and success rate. High-resolution representations (256×256 latent or full pixels) suffer from a critical domain-specific issue: **visual noise from rapid camera movements**.

In **Minecraft**, rapid view changes—such as fast camera rotations ("flicking")—are frequent. While these movements dramatically alter the pixel space (640×360), they often do not correspond to meaningful goal-oriented state transitions. Unlike the slow and continuous visual flow typically seen in robotics, these rapid shifts make dense pixel-level reconstruction noisy. This is reflected in the performance of Full Image Prediction (314 steps) and ESA Dense (313 steps), which lag behind our optimized method.

By compressing the visual input into a compact 8×8 latent grid, our ESA acts as a spatial low-pass filter. It extracts significant foreground objects (mobs, resources) while discarding high-frequency background noise, achieving the lowest trajectory length (293 steps) and a competitive success rate (32.5%). The 2×2 coarse grid remains close to the baseline (329 vs. 332 steps; 25.0% vs. 24.6% SR), indicating that excessive compression can discard useful spatial context.

J.3 Resolution across domains

MC uses MC-trained models; the other columns use PE-trained models with no GTA V or GameWorld fine-tuning. MC, PE, and GTA V report mean$\pm$std over 100 evaluation runs; GameWorld is a state-verifiable point estimate. The MC column is the enhanced ESA-only sweep (32.5% at 8×8; default ESA-only: 30.6%), whereas PE uses the full model (66.4%). These columns are separate configurations, not an identically controlled pooled ablation.

The 8×8 grid has the highest reported mean in each tested setting, supporting this choice here rather than universal optimality. ESA is an auxiliary target: the policy still receives original RGB for fine detail.

Table 18: ESA resolution across evaluation settings. MC uses MC-trained models; the remaining columns use PE-trained models. GameWorld is a point estimate.

Resolution	MC SR (%)	PE SR (%)	GTA V Kill SR (%)	GameWorld Simulation SR (%)
No ESA	24.6 ± 1.8	53.4 ± 1.4	6.6 ± 0.8	5.0
2×2	25.0 ± 2.1	57.1 ± 1.7	12.2 ± 1.8	15.0
8×8	32.5 ± 1.6	66.4 ± 1.2	19.1 ± 1.2	20.0
64×64	30.8 ± 1.8	62.9 ± 1.4	17.5 ± 1.5	15.0
256×256	28.6 ± 2.0	61.4 ± 1.6	16.3 ± 1.7	10.0

MC reproduces the enhanced ESA-only sweep; PE uses the full-model configuration; GTA V and GameWorld use PE-trained models without target-domain adaptation. These columns are not pooled into one cross-domain mean.

J.4 Grid versus token-matched bounding boxes

BBox-ESA serializes connected components from the same annotations as category labels and normalized coordinates, ordered by relevance and length-matched to the same 64-token budget. This uses the enhanced MC ESA-only configuration.

Table 19: ESA target representation at the same 64-token budget. The grid result corresponds to the enhanced ESA setting.

ESA target	MC SR (%)
8×8 grid	32.5 ± 1.6
Token-matched BBox-ESA	26.7 ± 2.0

The grid reaches $32.5 \pm 1.6\%$ versus $26.7 \pm 2.0\%$ for BBox-ESA. It covers foreground and background structure, whereas selected boxes can omit relevant regions. A 2D semantic grid still lacks height, collision geometry, and calibrated traversability.

K Cross-game transfer

K.1 Quantitative Hit/Kill evaluation

All policies in Tables 20 and 21 are PE-combat-trained and evaluated without target-game fine-tuning or adaptation. Values are mean±std over 100 evaluation runs with paired manifests and the same horizons, targets, loadouts, and verifiers across methods. Hit requires at least one valid strike; Kill requires defeating the designated target. External-baseline versions are specified in Appendix N.2.

Table 20: Hit success rates (%, mean ± std) across all six target games.

Method	Valorant	Minecraft	CS2	GTA V	Elden Ring	Mount & Blade
Vanilla IL	27.1 ± 1.8	8.6 ± 0.9	18.6 ± 1.5	15.7 ± 1.3	7.1 ± 0.8	8.6 ± 0.9
IL + CoT + Caption	35.1 ± 2.0	17.7 ± 1.2	28.3 ± 1.7	26.0 ± 1.6	13.2 ± 1.0	16.1 ± 1.1
CombatVLA	31.5 ± 1.9	13.6 ± 1.0	23.9 ± 1.6	21.4 ± 1.4	10.5 ± 0.9	12.7 ± 1.0
Open-P2P	39.4 ± 2.2	22.1 ± 1.4	35.1 ± 1.9	32.9 ± 1.8	17.7 ± 1.1	20.6 ± 1.2
3A-VLA	42.9 ± 2.3	25.7 ± 1.5	40.7 ± 2.1	38.6 ± 2.0	21.4 ± 1.3	24.3 ± 1.4

These results support zero-shot cross-game combat transfer under this protocol. Kill SR remains 10.6–21.4% for 3A-VLA, so the results do not establish broad game competence. Foundation-model pretraining exposure to these titles cannot be excluded.

K.2 Opposing instructions and chance controls

We evaluate PE-trained models on MC without fine-tuning. For each of 200 restored initial RGB states per conflict type, two separate rollouts execute opposing instructions: cover/mine a wall,

Table 21: Kill success rates (%, mean ± std) across all six target games.

Method	Valorant	Minecraft	CS2	GTA V	Elden Ring	Mount & Blade
Vanilla IL	12.2 ± 1.1	3.4 ± 0.6	8.4 ± 0.9	6.6 ± 0.8	2.8 ± 0.5	3.6 ± 0.6
IL + CoT + Caption	16.5 ± 1.2	7.8 ± 0.7	13.0 ± 1.0	11.4 ± 1.0	5.7 ± 0.6	7.1 ± 0.7
CombatVLA	14.8 ± 1.1	6.0 ± 0.6	11.0 ± 1.0	9.2 ± 0.8	4.4 ± 0.5	5.5 ± 0.6
Open-P2P	18.9 ± 1.3	10.4 ± 0.8	16.8 ± 1.1	15.1 ± 1.1	8.0 ± 0.7	9.5 ± 0.7
3A-VLA	21.4 ± 1.4	12.8 ± 0.9	20.2 ± 1.3	19.1 ± 1.2	10.6 ± 0.8	12.0 ± 0.8

enter/leave a doorway, or engage/evade an enemy. Directional SR measures each complete rollout. Both-Correct directly measures the states for which both rollouts succeed, rather than multiplying directional SRs. Uniform random valid actions test the complete task. An execution-matched control freezes PE-trained 3A-VLA and independently randomizes the A/B instruction for each target rollout while keeping the target success criterion fixed. The MC-trained row is an in-domain reference.

Table 22: Wall conflicts: directional and Both-Correct success rates (%). The PE-trained variants are evaluated without MC adaptation.

Method	Cover	Mine	Both-Correct
MC in-domain reference	31.1 ± 3.0	66.1 ± 3.2	24.2 ± 3.0
Uniform random actions	0	0	0
Random instruction	37.3 ± 3.6	6.8 ± 1.0	2.3 ± 0.7
ESA only	43.9 ± 3.3	9.1 ± 0.9	3.0 ± 0.8
IA+ESA, no AAR	47.6 ± 3.0	12.2 ± 1.2	5.9 ± 1.0
3A-VLA (PE→MC)	52.4 ± 2.8	15.6 ± 1.3	7.5 ± 1.1

Table 23: Doorway conflicts: directional and Both-Correct success rates (%). The PE-trained variants are evaluated without MC adaptation.

Method	Enter	Leave	Both-Correct
MC in-domain reference	71.2 ± 3.4	48.6 ± 3.5	30.0 ± 3.4
Uniform random actions	0	0	0
Random instruction	39.6 ± 3.5	10.1 ± 1.7	4.2 ± 0.9
ESA only	52.3 ± 3.8	10.7 ± 1.5	6.6 ± 1.1
IA+ESA, no AAR	59.7 ± 3.5	19.6 ± 2.0	11.8 ± 1.4
3A-VLA (PE→MC)	63.8 ± 3.3	24.7 ± 2.2	14.3 ± 1.5

Table 24: Enemy conflicts: directional and Both-Correct success rates (%). The PE-trained variants are evaluated without MC adaptation.

Method	Engage	Evade	Both-Correct
MC in-domain reference	80.1 ± 3.2	47.2 ± 3.6	32.5 ± 3.5
Uniform random actions	0	0	0
Random instruction	43.2 ± 3.6	14.4 ± 1.8	5.1 ± 1.0
ESA only	57.6 ± 3.7	15.2 ± 1.8	8.9 ± 1.2
IA+ESA, no AAR	64.8 ± 3.4	23.8 ± 2.2	14.9 ± 1.5
3A-VLA (PE→MC)	68.9 ± 3.1	27.4 ± 2.3	17.8 ± 1.6

Random instruction obtains $2.3/4.2/5.1\%$ Both-Correct for wall/doorway/enemy versus $7.5/14.3/17.8\%$ for 3A-VLA. A 25% binary-selection rate is not the appropriate comparator because two full executions must succeed. IA and AAR improve the reported means, supporting partial cross-domain, instruction-conditioned disambiguation; no per-conflict significance is claimed. Directional asymmetry remains, particularly for mining (15.6%), and the MC reference remains stronger.

K.3 Qualitative examples

Figure 9 retains representative deployment examples in GTA V, Elden Ring, and Mount & Blade without target-game fine-tuning. They illustrate short-horizon behaviors under different styles/interfaces; quantitative combat evidence is reported separately in Tables 20 and 21.



(a) GTA V: driving and on-foot navigation in an unseen urban environment.



(b) Elden Ring: third-person exploration and combat with dramatically different visual style.



(c) Mount & Blade: medieval combat with horseback control, entirely absent from training.

Figure 9: **Qualitative deployment examples in three target games.** The images illustrate short-horizon driving (a), exploration (b), and combat (c). They do not establish whole-game competence or exclude foundation-model pretraining exposure. Quantitative combat transfer is evaluated separately in Tables 20 and 21.

What works. Short-horizon behaviors can reuse IA primitives and coarse ESA categories such as person, vehicle, and cover. The observations are consistent with reusable cues, without proving fully domain-invariant semantics or an explicit traversability representation.

What breaks. Long-horizon game-specific mechanics (weapon-switching menus in Elden Ring, horseback combat in Mount & Blade, vehicle physics in GTA V) fall outside both IA's keyword coverage and the training distribution. The Hit/Kill protocol evaluates specified short-horizon combat tasks, not these broader game mechanics. Low absolute Kill rates and the opposing-instruction gap delimit the demonstrated transfer.

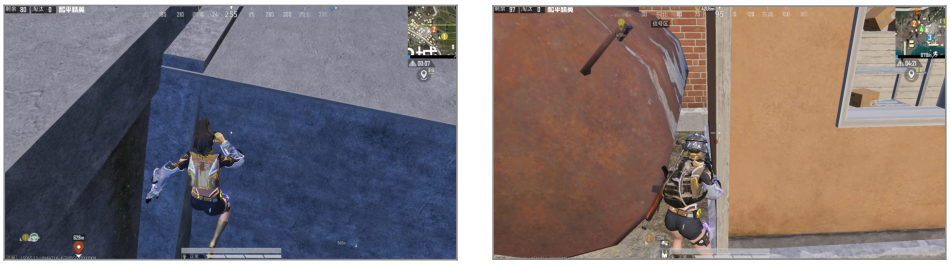
L Failure analysis

To investigate the limitations of the two abstraction channels, we analyze failed episodes under the enhanced IA+ESA configuration. This variant employs stronger IA annotations and a finetuned ESA segmenter, achieving an average success rate of 67.9% across six Peacekeeper Elite tasks. Figure 10 illustrates representative failures, while Table 25 summarizes the per-task results and manually attributed failure categories.

(A) Perception and semantic grounding failures — target recognition and interaction (IA-dominant). Combat Engagement exhibits the lowest success rate (61.4%), with 75.3% of failures attributed to IA-related causes. Partially occluded or distant opponents are particularly challenging to



(a) Failure cases in perception: partially occluded enemies and small-scale distant targets are more likely to be missed.



(b) Failure cases in navigation: the agent may become trapped on challenging terrain or in narrow spaces due to limited traversability estimation and collision handling.

Figure 10: Representative failure modes of the enhanced IA+ESA variant in Peacekeeper Elite.

Table 25: **Per-task failure analysis in Peacekeeper Elite (enhanced IA+ESA, without AAR).** Results are obtained with inference-time token pruning. IA-related and ESA-related percentages indicate the proportions of failed episodes attributed to each category, with dominant failure modes identified by manual inspection.

Task	**SR (%)**	**IA fail (%)**	**ESA fail (%)**	**Dominant mode**
Teammate Revival	74.3	68.2	31.8	Logic
Precision Parachuting	71.4	22.5	77.5	Control (overshoot)
Strategic Rotation	68.6	27.4	72.6	Navigation (B)
Vehicle Acquisition	67.1	72.7	27.3	Visual search
Resource Scavenging	64.3	33.6	66.4	Navigation (B)
Combat Engagement	61.4	75.3	24.7	Perception (A)

identify, leading to missed engagements or incorrect target selection. Similarly, Vehicle Acquisition (67.1%, 72.7% IA failures) and Teammate Revival (74.3%, 68.2% IA failures) rely on accurately identifying task-relevant entities and selecting appropriate interactions. These failures highlight limitations in visual target recognition and semantic grounding, particularly when relevant targets are small, occluded, or outside the immediate field of view.

(B) Navigation and spatial control failures — terrain and narrow-gap entrapment (ESA-dominant). Resource Scavenging (64.3%, 66.4% ESA failures), Strategic Rotation (68.6%, 72.6% ESA failures), and Precision Parachuting (71.4%, 77.5% ESA failures) are primarily limited by spatial control and navigation. Although ESA provides compact spatial affordances, its coarse representation does not explicitly encode terrain height, collision boundaries, or traversability. Consequently, the agent may become trapped on uneven terrain or in narrow passages, preventing task completion. In Precision Parachuting, limited spatial precision can also result in landing overshoot near small targets.

Implications for future work. The failure attribution highlights complementary limitations of the two abstraction channels. IA-dominant failures motivate more reliable object-level semantic grounding and target recognition, particularly under occlusion and partial visibility. ESA-dominant failures motivate finer spatial representations and explicit modeling of occupancy, traversability, and terrain geometry. Adaptive spatial resolution may further improve local precision while preserving the robustness of compact environment abstractions.

M Additional Results Across Backbones and Environments

M.1 Backbone scaling

Section 4.5 presents the component-wise cost–performance analysis. Table 26 provides complementary results across two VLM backbones and two environments. For each backbone, we compare the imitation-learning baseline without IA, ESA, or AAR against 3A-VLA.

Table 26: **Cost-effectiveness analysis across Minecraft and Peacekeeper Elite.** Comparison between Qwen2-VL and Qwen3-VL baselines versus our 3A-VLA approach.

Tier	Configuration	MC SR (%)	Peacekeeper Elite SR (%)
Baseline	Qwen2-VL-7B (No IA, no ESA, no AAR)	24.6	53.4
Ours	**Qwen2-VL-7B 3A-VLA**	**40.4**	**66.4**
Baseline	Qwen3-VL-8B (No IA, no ESA, no AAR)	33.9	58.8
Ours	**Qwen3-VL-8B 3A-VLA**	**50.1**	**69.3**

M.2 GameVerse: complementary task-specific transfer

We use the official GameVerse [39] no-reflection protocol. Forza Horizon 5 tests vehicle control; RDR2 tests navigation, combat, and interaction. CombatVLA, Open-P2P, and 3A-VLA use the same PE training data without GameVerse-specific fine-tuning or adaptation. Results use three trials per setting; Qwen3-VL-8B is the published benchmark reference.

Table 27: Complementary GameVerse evaluation under the no-reflection protocol.

Method	FH5 score	RDR2 score
Qwen3-VL-8B	3.2 ± 0.0	7.2 ± 5.4
GPT-4o	2.4 ± 2.2	10.0 ± 0.0
CombatVLA	3.2 ± 0.0	7.2 ± 5.4
Open-P2P	3.2 ± 0.0	7.2 ± 5.4
3A-VLA	5.4 ± 2.1	10.0 ± 0.0

Coarse progress events can map distinct rollouts to identical aggregate scores; ties do not establish statistical equivalence.

Table 28: GameVerse trial-level progress events for tied aggregate scores.

Method	FH5 checkpoints	RDR2 milestones	FH5 / RDR2
Qwen3-VL-8B	[1, 1, 1]	[1, 1, 0]	3.2 ± 0.0 / 7.2 ± 5.4
CombatVLA	[1, 1, 1]	[0, 1, 1]	3.2 ± 0.0 / 7.2 ± 5.4
Open-P2P	[1, 1, 1]	[1, 1, 0]	3.2 ± 0.0 / 7.2 ± 5.4

Equal scores for Qwen3-VL-8B, CombatVLA, and Open-P2P are not copied trajectories. The evaluator records discrete progress events: all pass one FH5 checkpoint per trial; in RDR2, each completes the first milestone in two of three trials, sometimes in different trial orders (Table 28). Under the benchmark's reporting convention, these event distributions map to the same normalized means and standard deviations. Ties do not distinguish behavior within a score bin or establish statistical significance/stability. 3A-VLA ties GPT-4o on RDR2.

M.3 GameWorld: state-verifiable simulation tasks

For GameWorld [23] Simulation, SR and normalized progress (PG) are evaluated from serialized game states rather than an image judge. CombatVLA, Open-P2P, and 3A-VLA use PE training without target-benchmark fine-tuning or adaptation.

Table 29: Complementary GameWorld Simulation evaluation. SR and normalized progress are verified from serialized states.

Method	SR (%)	Progress
Claude-Sonnet-4.6	15.0	16.6
Gemini-3-Flash-Preview	10.0	21.1
GPT-5.2	10.0	16.9
Qwen3-VL-235B-A22B	0.0	3.6
CombatVLA	0.0	3.4
Open-P2P	5.0	12.4
3A-VLA	20.0	22.3

These are reported point estimates; no uncertainty estimates are available.

GameWorld values are point estimates without reported variance. Together, these benchmarks provide complementary evidence on the tested control tasks. We describe the results as competitive among the evaluated baselines under these protocols, rather than statistically established superiority, universal game competence, or long-horizon planning.

N Training, evaluation, and deployment details

N.1 Training setup

AAR scorer implementation details. The scorer is intentionally small relative to the Qwen2-VL-7B backbone: for $d = 3584$ and $d_s = 512$, it adds fewer than 0.2% parameters while still allowing nonlinear IA–ESA interactions. Each residual MLP block maps $\mathbb{R}^{d_s} \to \mathbb{R}^{d_s}$ as

$$\mathrm{FFN}(\mathbf{z}) = W_o\big(\mathrm{SiLU}(W_g\mathbf{z}) \odot W_v\mathbf{z}\big), \qquad \mathbf{z} \leftarrow \mathrm{LN}\big(\mathbf{z} + \mathrm{Dropout}(\mathrm{FFN}(\mathbf{z}))\big),$$

where W_g and W_v expand to $4d_s$ channels and W_o projects back to d_s. We apply no sigmoid or clipping to q_i; its dynamic range is controlled by the batch softmax defined in Section 3.4 and the entropy regularizer $\lambda_{\bar{H}} = 0.01$. In distributed training, the scalar scores are normalized over the synchronized per-forward global mini-batch (8 samples in our setting with local batch 1 on 8 GPUs); gradient accumulation only changes the optimizer batch size. This synchronized normalization avoids the degenerate local-batch case where $B = 1$ would yield $w_i = 1$ and disable AAR. The detailed architecture of the AAR scorer is summarized in Table 30.

Causal masking and inference. The training sequence places abstraction tokens *after* the action chunk, so action decoding is causally shielded from IA/ESA during both training and inference. At deployment we decode only the action chunk; IA and ESA contribute zero latency.

Table 30: **Architecture of the AAR scorer g_ψ.**

Component	Configuration
Input Layer	
Input feature	Relation feature $\mathbf{r}_i \in \mathbb{R}^d$
Bottleneck projection	Linear($d \to d_s$) + GELU + LayerNorm
Bottleneck dim	$d_s = 512$
MLP Encoder	
Number of layers	2 residual MLP layers with LayerNorm
Hidden dim	$4d_s$ with SwiGLU activation
Dropout	0.1
Projection Head	
Score projection	Linear($d_s \to d_s/2$) $\to$ GELU $\to$ Linear($d_s/2 \to 1$)
Output	Scalar score $q_i \in \mathbb{R}$

Table 31: **Training hyperparameters.**

Parameter	Value
Base model	Qwen2-VL-7B (full SFT, separate per game)
Training loss	Autoregressive CE on [action][IA][ESA] jointly
Optimizer	AdamW, weight decay 0.01, bf16
Per-device batch / Grad accum	1 / 2 (effective batch 16 on 8 GPUs)
LR / Schedule / Warmup	1e-5 / cosine / 3%
Epochs	1
Train hardware	8×H20 GPUs
Input resolution	640×360
Data	50h Minecraft, 200h Peacekeeper Elite
Training sequence	[visual][instruction]→[action][IA][ESA]
Inference sequence	[top-k visual][instruction]→[action] (IA/ESA truncated)

Latency breakdown. The default action-first model uses 1.1 s per input-to-action step on a single L40S. The separate pruned deployment uses approximately 0.3 s: capture 15 ms, ViT 45 ms, pruning 5 ms, LLM decode 225 ms, and post-processing 10 ms. This breakdown describes the pruning configuration in Section 4.6, not the unpruned primary benchmark model.

N.2 External baselines and evaluation uncertainty

At the time of the rebuttal evaluation, the requested CombatVLA, Lumine, and Game-TARS trained checkpoints were unavailable. We reproduce CombatVLA with a 7B model under the same PE data and budget and adapt the public Open-P2P checkpoint [36] under the same data, budget, and interface. Throughout the added evaluations, CombatVLA denotes our reproduction and Open-P2P the adapted model. They are external-reference baselines, not capacity-matched controls; the Qwen2-VL IL variants retain the same backbone and data as 3A-VLA. Table 32 reports mean SR and standard deviations for the five methods in Table 2 with available run-to-run statistics from 100 evaluation runs using shared episode manifests. The standard deviations quantify variability across evaluation runs for the trained policies. The table reports the standard deviation of the aggregate SR separately. The PE combat difference is 20.3 percentage points ($67.1 \pm 3.6\%$ versus $46.8 \pm 5.0\%$ for IL+CoT+Caption); marginal deviations alone do not establish significance.

For MCU, the external-reference models use the same MC data and budget. Table 33 gives added Mini comparisons; no added All-protocol numbers were reported. The 40.4% MC ablation result is the macro-average of the three MCU Mini groups, $(40.6 + 45.9 + 34.7)/3$. The enhanced annotation variant is reported separately.

Table 32: Peacekeeper Elite success rates (%, mean ± standard deviation) over 100 evaluation runs with shared episode manifests.

Method	Parachuting	Scavenging	Combat	Revival	Vehicle	Rotation	Avg.
Vanilla IL	54.3 ± 2.7	48.6 ± 3.7	44.3 ± 4.4	68.6 ± 2.3	51.4 ± 3.9	52.9 ± 3.1	53.4 ± 1.4
IL + CoT + Caption	56.3 ± 3.1	50.9 ± 4.2	46.8 ± 5.0	69.3 ± 2.7	53.7 ± 4.4	55.0 ± 3.6	55.3 ± 1.6
CombatVLA	48.1 ± 3.5	44.7 ± 4.4	42.1 ± 4.2	44.0 ± 3.2	47.1 ± 4.2	49.0 ± 3.8	45.8 ± 1.6
Open-P2P	48.7 ± 3.8	45.2 ± 4.6	63.5 ± 5.4	54.6 ± 4.1	58.3 ± 4.7	60.1 ± 3.5	55.1 ± 1.8
3A-VLA	66.1 ± 2.4	60.5 ± 3.5	67.1 ± 3.6	70.2 ± 2.0	68.9 ± 3.1	65.7 ± 2.7	66.4 ± 1.2

Table 33: Additional Minecraft comparisons, including the enhanced annotation variant.

Method	Embodied Steps	Embodied SR	Combat Steps	Combat SR	GUI Steps	GUI SR
CombatVLA	310	29.5 ± 4.8	347	20.1 ± 3.3	343	24.0 ± 4.3
Open-P2P	297	34.0 ± 5.3	329	31.7 ± 4.8	326	29.5 ± 4.6
3A-VLA	278	40.6 ± 3.2	271	45.9 ± 3.5	307	34.7 ± 2.7
3A-VLA (Enhanced)	252	45.0 ± 3.0	248	50.7 ± 3.2	301	39.3 ± 2.5

N.3 Output order and input-to-action latency

We change only target serialization order and use the same single-L40S input-to-action timing boundary over 100 evaluation runs. Action-first decoding cannot attend to future IA/ESA tokens and stops after the action chunk.

Table 34: Output-order comparison. End-to-end input-to-action latency is measured on a single L40S.

Output order	MC SR (%)	Latency (s)
[action][IA][ESA]	40.4 ± 1.5	1.1
[IA][action][ESA]	39.9 ± 1.7	1.3
[IA][ESA][action]	37.6 ± 1.8	3.2

IA-first changes MC SR from 40.4% to 39.9% and latency from 1.1 to 1.3 s; placing both auxiliary spans first yields 37.6% at 3.2 s. Explicitly decoding the tested abstractions before acting adds latency without improving mean SR.

N.4 Action-switching frequency

Switches/100 counts changes between adjacent complete action vectors per 100 transitions. A–B–A reversals/100 counts three-step oscillations whose endpoints match and differ from the middle action, normalized per 100 eligible three-step windows. All methods use the same action-vector representation and evaluation interface.

Table 35: Action-switching frequency and success rate on Peacekeeper Elite.

Method	Switches/100	A-B-A/100	PE SR (%)
Vanilla IL	42.6 ± 10.8	9.8 ± 2.5	53.4 ± 1.4
IL + CoT + Caption	11.2 ± 0.7	2.6 ± 0.3	55.3 ± 1.6
CombatVLA	25.4 ± 4.1	5.9 ± 1.2	45.8 ± 1.6
Open-P2P	16.7 ± 1.8	3.8 ± 0.6	55.1 ± 1.8
3A-VLA	7.8 ± 0.6	1.2 ± 0.2	66.4 ± 1.2

3A-VLA has fewer switches and reversals while retaining the highest mean PE SR among these methods. This addresses observed instability; the association does not establish that reduced switching causes the SR gain.

O Ethical considerations

Automated game aiming. We directly address the concern on automated aiming assistance. (1) *Coarse action space*: 3A-VLA uses discretized yaw/pitch bins and keyboard/mouse controls without rendering-pipeline hooks. Discretization reduces target-snapping precision but does not by itself prevent automated-aiming misuse. (2) *Latency*: The default input-to-action latency is 1.1 s; the pruned configuration takes approximately 0.3 s. These measurements are not a general guarantee against misuse. (3) *Access control*: To mitigate potential misuse and address data governance requirements, the Peacekeeper Elite dataset and model checkpoints are hosted on Hugging Face with gated access. Any granted access is subject to research-use restrictions and explicit prohibitions against commercial exploitation, competitive deployment, or adversarial use.

Broader dual-use considerations. Any agent that acts autonomously in interactive environments carries dual-use risk. We identify two additional vectors beyond aiming: (i) *economy manipulation*—an autonomous scavenger could, in principle, farm in-game resources at scale; and (ii) *social deception*—an agent controlling an avatar could impersonate a human teammate. We address these risks through the release conditions below and by not training or releasing social interaction modules (text chat, voice, emotes). Coarse actions and latency alone are insufficient safeguards.

Data and privacy. The PE trajectories were produced by consenting internal full-time collection/annotation staff in operator-authorized dedicated test sessions under the stated IRB-approved protocol. Staff were informed of the research purpose, recorded gameplay/controls, data handling, and intended use before collection; their work was compensated through regular employment arrangements. Minecraft instead follows the public VPT source and applicable terms (Appendix A); the PE staff protocol is not attributed to third-party VPT contributors. Before PE training and release, structured and visible identifiers, including names and kill-feed actors, are masked. Released processed artifacts exclude identifiers and chat/voice data.

Staff collection instructions. Internal staff were assigned the following collection task brief:

> Use the assigned research account in an authorized dedicated test session. Follow the supplied task instruction using ordinary controls: land at the designated waypoint, collect the specified supplies, engage the designated controlled target, revive the designated teammate, locate and board an available vehicle, or move toward the specified safe zone. Record the first-person gameplay and aligned controls required for the assigned task. Keep the session within the research environment; do not use public matchmaking or interact with uninformed players. Exclude voice/chat and personal identifiers from research artifacts, and apply identifier masking before training or release. Recordings are for the stated game-agent research under the consented collection protocol.

Collection and annotation were regular paid staff duties rather than marketplace crowdsourcing.

Experimental setting and benchmark rationale. All reported game evaluations used dedicated test environments, research accounts, and controlled targets, without public matchmaking or uninformed opponents. PE research use was operator-authorized; other games used lawfully obtained accounts or copies under applicable terms. Combat scenarios stress moving targets, occlusion, off-screen events, clutter, and short-horizon control, rather than violence as a research objective. Non-combat evaluation includes MCU tasks and five PE task families. Vehicle acquisition means boarding an available virtual vehicle. The experiments accessed no production game economies, platform back ends, physical actuators, or weapon interfaces.

Commercial assets and release boundaries. We claim no ownership or redistribution rights in commercial games, footage, or artwork. Coarse semantic maps do not remove IP obligations. The processed PE training release, PE framework, and gated finetuned checkpoints are linked through `https://3a-vla.github.io/`; excluded artifacts are source replays, unprocessed proprietary frames, raw replay/action logs, identifiers, chat/voice data, and commercial executables/assets. Reproducers must obtain lawful game access and follow platform terms. Proprietary execution conditions and limited coverage of players, regions, versions, and interfaces remain limitations.

Checkpoint release conditions. Finetuned checkpoints are released on Hugging Face with gated access. Access is granted only after privacy, IP, and security review, under a research-use-only license. The Hugging Face repositories include this LICENSE with a responsible-use addendum and list a security contact and a private reporting channel. The license prohibits competitive deployment, automated aiming, resource farming, economic manipulation, bulk-account automation, anti-cheat evasion, social impersonation, real-world weaponization, and unlawful tracking/surveillance.

Intended use and path forward. 3A-VLA is designed to study how embodied agents form sparse abstractions in complex 3D environments—a direction that may inform simulation, accessibility, and future embodied-control research. We evaluate virtual game interfaces and do not demonstrate physical-system transfer. We advocate for community norms analogous to those in penetration-testing research: *study the vulnerability, publish the mechanism, but do not ship the exploit*. To this end, we commit to (1) gated, reviewed access to finetuned checkpoints rather than unrestricted download; (2) a living responsible-use addendum in the repository that tracks emerging misuse vectors; and (3) collaboration with game platform trust-and-safety teams upon request.

P Matched abstraction-target controls

We compare a boundary-free unified target, two alternative two-span decompositions, and IA+ESA with the same Qwen2-VL-7B backbone, trajectories, semantic scope, total 100-token target budget, and auxiliary supervision, with AAR disabled. Organization controls add no rationales, retrieval, or loss terms. The three alternative annotations use the same Qwen2-VL-7B annotator and input: instruction plus sampled frames from a 4 s trajectory. All preserve goal/action primitives and observable entities, states, coarse spatial relations, topology, obstacles, operable regions, and supported interactions; only output organization changes. Complete control prompts follow. This target-organization study should be distinguished from the default SegFormer-derived ESA grid construction in Section 3.3.

Table 36: Abstraction organization with AAR disabled. The annotation study uses shared VLM control prompts, the same backbone, semantic scope, trajectories, and 100-token budget; the default SegFormer-grid construction is specified separately in Section 3.3.

Target organization	MC SR (%)	PE SR (%)
No abstraction	24.6 ± 1.8	53.4 ± 1.4
Image-caption only	27.0 ± 1.9	54.9 ± 1.6
Unified abstraction	32.4 ± 1.9	55.0 ± 1.5
Entity + Affordance	32.2 ± 1.8	56.8 ± 1.8
Where + How	33.7 ± 1.9	57.5 ± 1.6
IA + ESA	36.5 ± 1.7	62.9 ± 1.4

With AAR off, IA+ESA reaches $36.5/62.9\%$ MC/PE SR versus $32.4/55.0\%$ for Unified, $32.2/56.8\%$ for Entity+Affordance, and $33.7/57.5\%$ for Where+How. These comparisons support intention/environment separation within the tested controls, not a uniquely necessary decomposition for every domain. IA/ESA are jointly supervised auxiliary spans, not a mandated semantic reasoning order or an inference-time bottleneck.

P.1 Unified target

> System: You are an abstraction annotator for embodied game trajectories. Given first-person visual frames and the task instruction, extract both (i) the agent's current high-level goal and reusable goal/action primitives and (ii) observable entities, states, coarse spatial layout and topology, obstacles, operable regions, and supported interactions. Jointly emit these two information types in one mixed abstraction span. Do not separate them with labels, sections, special delimiters, or a prescribed Intention-first or Environment-first semantic order.
>
> Perform the intention and environment abstraction tasks jointly within this single unified target. Use observable evidence and avoid unsupported assumptions. Output only short, comma-separated primitives, with no full sentences or chain-of-thought. The entire output, including punctuation, has a hard maximum budget of 100 tokenizer tokens.

Input: [Instruction], [Sampled frames from 4s trajectory] Output: One boundary-free, comma-separated abstraction sequence only.

P.2 Entity and Affordance

System: You are an abstraction annotator for embodied game trajectories. Given first-person visual frames and the task instruction, extract both (i) the agent's current high-level goal and reusable goal/action primitives and (ii) observable entities, states, coarse spatial layout and topology, obstacles, operable regions, and supported interactions. Organize exactly the same information into two spans. [Entity] contains task-relevant entities and scene states with their coarse locations and relations. [Affordance] contains the goal-conditioned interactions and reusable goal/action primitives supported by those entities and states. Do not add information beyond these shared intention and environment requirements.

Use observable evidence and avoid unsupported assumptions. Output only short, comma-separated primitives, with no full sentences or chain-of-thought. The entire output, including both span labels and all punctuation, has a hard maximum budget of 100 tokenizer tokens.

Input: [Instruction], [Sampled frames from 4s trajectory] Output: [Entity] <comma-separated primitives> [Affordance] <comma-separated primitives>

P.3 Where and How

System: You are an abstraction annotator for embodied game trajectories. Given first-person visual frames and the task instruction, extract both (i) the agent's current high-level goal and reusable goal/action primitives and (ii) observable entities, states, coarse spatial layout and topology, obstacles, operable regions, and supported interactions. Organize exactly the same information into two spans. [Where] contains task-relevant entities, scene states, and their coarse locations, spatial relations, and topology. [How] contains the current high-level goal, reusable goal/action primitives, and goal-conditioned interactions supported by the observed environment. Do not add information beyond these shared intention and environment requirements.

Use observable evidence and avoid unsupported assumptions. Output only short, comma-separated primitives, with no full sentences or chain-of-thought. The entire output, including both span labels and all punctuation, has a hard maximum budget of 100 tokenizer tokens.

Input: [Instruction], [Sampled frames from 4s trajectory] Output: [Where] <comma-separated primitives> [How] <comma-separated primitives>

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: The main text scopes evidence to evaluated MC/PE baselines and zero-shot cross-game combat transfer under Hit/Kill evaluation. Matched target controls, AAR controls, opposing instructions, and complementary public benchmarks are in Appendices P, H.1, K.2, M.2, and M.3.

   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: Section 5 discusses the lack of explicit temporal memory and long-horizon planning, and limitations in terrain and collision geometry. Section 4.4 and Appendix K.1 discuss low absolute Kill rates and possible target-title exposure during pretraining. Appendices N and O explain proprietary PE execution and reproducibility constraints. Appendix L provides failure diagnostics.

   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: [N/A]

   Justification: The paper contains no theorem or theoretical performance guarantee. Equations define auxiliary targets, relation features, and the empirical objective. Appendix I.2 provides motivation/descriptive diagnostics rather than a proof.

   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: Section 4 and Appendix N disclose architecture, hyperparameters, hardware, data volumes, action interfaces, baseline adaptations, and evaluation protocols. Processed PE training data, the framework, and finetuned checkpoints are accessible through `https://3a-vla.github.io/`. Lawful game access and a compatible authorized environment are still required, limiting independent replication of every proprietary benchmark condition.

   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: [Yes]

   Justification: `https://3a-vla.github.io/` links processed PE training data, the runnable PE framework, and gated finetuned checkpoints. Appendix N specifies tools and exclusions. Source replays, unprocessed proprietary frames, raw replay/action logs, identifiers, chat/voice, and commercial assets are excluded. Checkpoint access is gated under research-use terms (Appendix O); open tools/data do not imply unrestricted access to proprietary environments.

   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: Section 4.1 and Appendix N describe benchmarks, metrics, training, baseline versions, paired evaluation, and latency. Appendices P and J.3 clarify matched controls and separate sweep configurations. No unreported All-protocol results or training-seed counts are inferred.

   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: The appendices report mean±standard deviation over 100 evaluation runs with paired manifests for PE where available (Table 32) and for six-game transfer. MCU retains reported uncertainty. These are evaluation-run deviations, not standard errors or 100 training seeds. Per-conflict significance is not claimed. GameWorld is a point estimate; GameVerse uses three trials and a coarse discrete metric (Appendices N.2, K.1, M.2, and M.3).

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: Section 4.1 and Appendix N specify 8 H20 training GPUs, single-L40S inference, 50h MC/200h PE data, and one epoch. Default 1.1 s timing is distinguished from the pruned 0.3 s breakdown. Section 4.5 and Appendices F and G give annotation GPU-hours, API costs, and person-hours.

   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: Appendix O documents controlled research environments, consented internal PE staff collection, privacy/IP boundaries, benchmark rationale, misuse risks, and gated release safeguards. Discretized actions and latency are not treated as sufficient guarantees against misuse.

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: Appendix O discusses potential research/accessibility benefits and misuse through aiming, farming, social impersonation, or harmful adaptation. Physical-system transfer is not evaluated. Research-use restrictions, gated checkpoint access, security reporting, and platform collaboration are described.

    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: Appendix O describes released checkpoints on Hugging Face with gated access, granted after privacy/IP/security review under a research-use-only LICENSE with a responsible-use addendum; the repositories list a security contact and private reporting channel. Raw proprietary/identifying artifacts are excluded.

    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: Existing models, VPT/MCU data, and baselines are cited. Appendix O distinguishes PE research authorization from redistribution rights, specifies lawful access/platform terms, and excludes commercial executables/assets. Coarse maps do not remove IP obligations.

    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: Processed PE training data, the PE framework, and finetuned checkpoints are linked through `https://3a-vla.github.io/`. Appendices B, C, F, and N describe collection, conversion, annotations, and evaluation; Appendix O states release boundaries.

    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: [Yes]

Justification: Appendix O provides the collection task brief and data-handling instructions. PE collection/annotation used internal full-time staff under regular paid employment arrangements, with informed consent rather than anonymous crowdsourcing. Appendix O describes consent/privacy boundaries.

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: [Yes]

    Justification: The stated IRB-approved protocol covers PE collection by consenting internal staff. Appendix O describes research purpose, recorded gameplay/controls, consent, controlled sessions, and privacy. This is scoped to PE staff, not the separate public VPT contributors.

    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: [Yes]

    Justification: LLMs/VLMs are a core component of the methodology. The paper clearly declares: (1) Qwen2-VL-7B is used as the base VLA model (Section 3.1), (2) Qwen2-VL-7B is used zero-shot for Intention Abstraction keyword generation (Section 3.2), and (3) GPT-4o with RAG is used as an enhanced IA annotator variant (Appendix F). Annotation and control prompts are provided in Appendices D, E, F, and P.

    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.