Controlling Multimodal Conversational Agents with Coverage-Enhanced Latent Actions

Anonymous ACL submission

Abstract

Vision-language models are increasingly employed as multimodal conversational agents (MCAs) for diverse conversational tasks. Recently, reinforcement learning (RL) has been widely explored for adapting MCAs to various human-AI interaction scenarios. Despite showing great enhancement in generalization performance, fine-tuning MCAs via RL still faces challenges in handling the extremely large text token space. To address this, we learn a compact latent action space for RL fine-tuning instead. Specifically, we adopt the learning from observation mechanism to construct the codebook for the latent action space, where future observations are leveraged to estimate current latent actions that could further be used to reconstruct future observations. However, the scarcity of paired image-text data hinders learning a codebook with sufficient coverage. Thus, we leverage both paired image-text data and text-only data to construct the latent action space, using a cross-modal projector for transforming text embeddings into image-text embeddings. We initialize the cross-modal projector on paired image-text data, and further train it on massive text-only data with a novel cycle consistency loss to enhance its robustness. We show that our latent action based method outperforms competitive baselines on two conversation tasks across various RL algorithms.

1 Introduction

Vision-language models (VLMs) (Yin et al., 2024) like Qwen-VL (Bai et al., 2025a) and GPT-4o (Hurst et al., 2024) are increasingly employed as multimodal conversational agents (MCAs) for various conversation tasks (Yao et al., 2025). MCAs enable emotionally rich and contextually grounded dialogues based on understanding both input images and texts, and thus become particularly valuable in fields like entertainment (Mehta et al., 2022), online education (Griol et al., 2014), and personalized assistants (Nguyen et al., 2024).

Recently, reinforcement learning (RL) (Sutton et al., 1998) has been widely explored for adapting MCAs to diverse real-world human-AI interaction scenarios (Zhou et al., 2025). Generally, RL algorithms frame response token generation in MCAs as a sequential decision-making process (Chen et al., 2021), which optimize the policy to maximize cumulative rewards through interacting with environments. Despite showing great enhancement in generalization performance (Chu et al., 2025), fine-tuning MCAs via RL still faces challenges in dealing with large exploration spaces. For instance, with token vocabulary size $|\mathcal{V}|$ and maximum response length m, the sampling space for RL scales exponentially as $|\mathcal{V}|^m$.

To address the challenge of large text token space, we learn a compact latent action space for RL fine-tuning instead, following previous works (Jia et al., 2025). Specifically, we adopt the learning from observation mechanism (Jiang et al., 2023; Ye et al., 2025) to construct the codebook for the latent action space, where future observations are leveraged to estimate current latent actions that could be further used to reconstruct future observations. As a result, the action sampling space at each step is reduced from the token vocabulary size $|\mathcal{V}|$ (e.g., 152K for Qwen2.5-VL (Bai et al., 2025b)) to the latent action codebook size $|\mathcal{C}|$ (e.g., 128).

Generally, the codebook has to be learned from diverse data with sufficient coverage, which is a prerequisite for effective RL exploration in latent spaces (Chen et al., 2025). Note that VLMs in MCAs are typically pre-trained on paired image-text corpora (V, T), which implicitly convey complementary and partially redundant information between visual and textual modalities (Radford et al., 2021). Unfortunately, while unpaired image collections and text corpora are abundant on the web, curating them into aligned image-text corpora remains prohibitively costly (Gupta et al., 2025), posing a dilemma in constructing latent spaces. On one

hand, using limited paired data and abundant unpaired data would introduce *unimodal bias* (Zhang et al., 2024a), where a model would overly rely on one modality and ignore others. On the other hand, training the codebook solely on limited paired data may result in insufficient coverage, thereby impairing the agent's generalization ability when handling diverse unseen conversation scenarios.

In this paper, we leverage both paired image-text data (V, T) and unpaired text-only data T to learn the coodbook for the latent space. To improve the coverage of latent actions while avoiding potentially unimodal bias, we attempt to construct pseudo paired data (V', T) based on text-only data T, and use the pseudo data (V', T) and the collected data (V, T) to learn the cookbook.

However, training a conditional image generator $G(V|T)$ for this purpose is computationally expensive due to the high dimension nature of images (Pope et al., 2021). Thus, we learn a cross-modal projector P instead, which transforms an input text e^T to an image-text pair $e^{V,T}$ in the embedding space, based on the cross-modal redundancy assumption (Radford et al., 2021). Concretely, for each item in the paired image-text data (V, T), we compute the text embedding e^T and image-text embedding $e^{V,T}$ using an existing encoder, and train the projector P to imitate the projection between these two kinds of embeddings. To enhance the robustness of the projector P, we further train it on massive text-only data T using a cycle consistency loss (Zhu et al., 2017). We introduce an additional projector P' that can transform image-text embedding $e^{V,T}$ back to text embedding e^T. In this way, we can optimize the projector P by enforcing cycle consistency on text-only data T such that $P'(P(e^T)) \approx e^T$.

We evaluate our method on two conversation tasks, namely multimodal role-playing conversation (Dai et al., 2025) and multimodal personalized conversation (Li et al., 2025a). To evaluate the generalizability of latent actions, we conduct experiments using various RL algorithms, such as GRPO (Shao et al., 2024) and Dr.GRPO (Liu et al., 2025). We construct the latent action space using paired image-text data (V, T) and text-only data T. The (V, T) data are comprised of image-caption pairs, multimodal news articles, and multimodal Wikipedia pages, totaling 14M images and 1B text tokens. The text-only data are mainly derived from SlimPajama (Soboleva et al., 2023), which contains 627B text tokens. Experimental results show that our method outperforms competitive baselines.

In summary, our work makes the following three key contributions. 1) We are the first to introduce latent actions for fine-tuning multimodal conversational agents via RL, which significantly reduces the exploration space. 2) We construct the latent action space with both paired image-text data and text-only data, using a cross-modal projector trained with a novel cycle consistency loss. 3) We evaluate our latent action based method on two multimodal conversation tasks and demonstrate that our method outperforms competitive baselines, and further show that the cross-modal projector is critical for improving the coverage of latent actions.

2 Preliminary

Reinforcement Learning for VLM Agents In reinforcement learning (RL), problems are framed by a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$. For VLMs, the state at step t is the contextual information $s_t = (x^V, x^{T_{1:t}}) \in \mathcal{S}$, which includes the input image x^V and the current token sequence $x^{T_{1:t}}$. $\mathcal{A}$ is the action space containing all possible actions a_t at each step. $\mathcal{T}$ is the state transition function, governing the transition from s_t to s_{t+1}, i.e., $P\big(s_{t+1} \mid s_t, a_t\big)$. The reward function $\mathcal{R}\big(x^{T_{p+1:m}}\big)$ assigns a scalar reward to the response $x^{T_{p+1:m}}$, conditioned on the input $(x^V, x^{T_{1:p}})$, with prompt length p and maximum sequence length m, following common practice in RL for VLMs (Shen et al., 2025).

Latent Actions for Reinforcement Learning In traditional token-level RL, each action a_t corresponds to selecting the next text token $x^{T_{t+1}}$ from the token vocabulary $\mathcal{V}$, i.e., $\mathcal{A} = \mathcal{V}$. While in latent action RL, at each step t, the policy $\pi_\theta(a_t|x^V, x^{T_{1:t}})$ selects a latent action a_t from a compact codebook $\mathcal{C}$, i.e., $\mathcal{A} = \mathcal{C}$. During RL exploration, the latent action policy samples a latent action at each step, ultimately yielding the terminal state s_m. During exploitation, the latent action policy is refined to maximize expected rewards using RL algorithms such as GRPO (Shao et al., 2024).

3 Methodology

In this section, we first describe the overall model design for incorporating latent actions into VLMs (Sec. 3.1). Next, we detail the unsupervised construction of the latent action space (Sec. 3.2). Finally, we introduce the procedure of latent action based RL fine-tuning (Sec. 3.3).

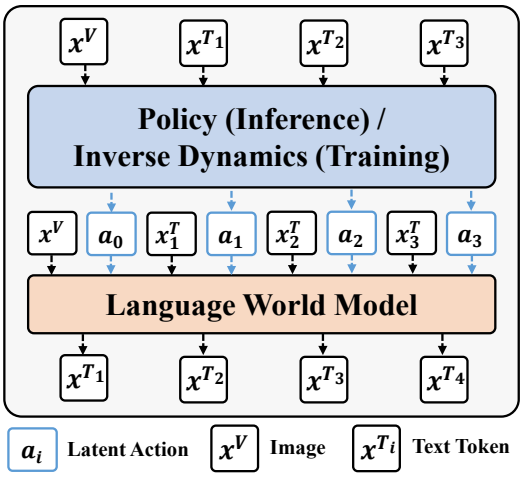


Figure 1: Illustrations of integrating latent actions with vision-language models.

3.1 Model Design

To fine-tune MCAs via latent action RL, we introduce three new modules, as illustrated in Figure 1. These modules are designed to share a base VLM while adding a small number of additional parameters, thereby introducing only marginal computational overhead. For further details on the model design, please refer to the Appendix A.

Language World Model f_{world} The language world model $f_{\text{world}}(x^{T_{t+1}}|x^V, x^{T_{1:t}}, a_t)$ takes current observations $(x^V, x^{T_{1:t}})$ and a latent action a_t as input, and auto-regressively outputs the next token $x^{T_{t+1}}$. The latent action a_t is provided by the inverse dynamics model f_{inverse} during constructing the latent action space, and by the policy π_θ during inference and RL phases.

Inverse Dynamics Model f_{inverse} The inverse dynamics model $f_{\text{inverse}}(a_t|x^V, x^{T_{1:t+1}})$ takes future observations $(x^V, x^{T_{1:t+1}})$ as input, and outputs a discrete latent action index $a_t \in \{1, \ldots, |\mathcal{C}|\}$ for the current step. The corresponding latent action embedding $c_{a_t} = \mathcal{C}[a_t] \in \mathbb{R}^d$ is then retrieved from the trainable codebook $\mathcal{C} \in \mathbb{R}^{|\mathcal{C}|\times d}$ and used by f_{world} to reconstruct the next token $x^{T_{t+1}}$. Note that f_{inverse} only assists training and does not serve for the inference phase.

Policy Model π_θ The latent action policy model $\pi_\theta(a_t|x^V, x^{T_{1:t}})$ takes the current observations $(x^V, x^{T_{1:t}})$ as input, and predicts latent action a_t for the current step. Since the language world model f_{world} is controlled by latent actions, we can optimize the latent action distribution of π_θ for steering f_{world} to generate responses toward higher rewards.

3.2 Latent Action Space Learning

Following Jia et al. (2025), we construct the latent action space using large-scale corpora in two steps. 1) *inverse dynamics learning*, which trains the f_{world}, f_{inverse}, and $\mathcal{C}$ in an unsupervised manner (Fig. 2 (a)); 2) *policy behavior cloning*, which trains the policy model π_θ to mimic the latent action a_t inferred by f_{inverse} (Fig. 2 (b)).

3.2.1 Inverse Dynamics Learning

We first outline the overall objective of inverse dynamics learning, followed by the training procedure of the introduced cross-modal projector.

Overview As shown in Fig. 2 (a), we jointly train the inverse dynamics model f_{inverse}, language world model f_{world}, and the latent action codebook $\mathcal{C}$, using the mixed corpus $\mathcal{D}^{VT} \cup \mathcal{D}^T$ (paired image-text data and text-only data). The loss is as:

$$\mathcal{L}_{\text{inverse}} = \mathbb{E}_{\mathcal{D}^{VT}\cup\mathcal{D}^T}\left[-\sum_{t=1}^{m-1} \log f_{\text{world}}(x^{T_{t+1}}|e_t^{V,T}, a_t)\right], \tag{1}$$

where the expectation is taken over sequences $(x^V, x^{T_{1:m}})$ sampled from the mixed corpus $\mathcal{D}^{VT} \cup \mathcal{D}^T$, with $a_t = f_{\text{inverse}}(e_{t+1}^{V,T}) \in \{1, ..., |\mathcal{C}|\}$. The embedding $e_t^{V,T}$ is obtained via:

$$e_t^{V,T} = \begin{cases} f_{\text{VLM}}(x^V, x^{T_{1:t}}), & \text{if } x^V \neq \emptyset \quad (\text{from } \mathcal{D}^{VT}); \\ P(f_{\text{VLM}}(x^{T_{1:t}})), & \text{if } x^V = \emptyset \quad (\text{from } \mathcal{D}^T), \end{cases} \tag{2}$$

where f_{VLM} denotes the encoding module based on VLMs. P denotes the cross-modal projector for transforming text embeddings into image-text embeddings, and its training procedure is as follows.

Cross-modal Projector Training Let P denote the forward cross-modal projector, which maps text embeddings e_t^T to the parameters of a diagonal Gaussian distribution over the image-text embedding space, i.e., $(\mu_t, \sigma_t) = P(e_t^T)$. Let P' denote the reverse projector, which maps image-text embeddings back to the text embedding space. We train P and P' in the following two steps.

Step 1: Initialization on paired image-text data. We first train the forward projector P on paired image-text data $\mathcal{D}^{VT}$, where the loss is defined as:

$$\mathcal{L}_{\text{t2vt}} = \mathbb{E}_{\mathcal{D}^{VT}}\left[\sum_{t=1}^{m-1} \frac{1}{2}\left(\left\|\frac{e_t^{V,T} - \mu_t}{\sigma_t}\right\|^2 + \|\log \sigma_t^2\|_1\right)\right], \tag{3}$$

where the expectation is taken over sequences $(x^V, x^{T_{1:m}}) \sim \mathcal{D}^{VT}$, and $e_t^{V,T} = f_{\text{VLM}}(x^V, x^{T_{1:t}})$, and $(\mu_t, \sigma_t) = P(e_t^T = f_{\text{VLM}}(x^{T_{1:t}}))$.

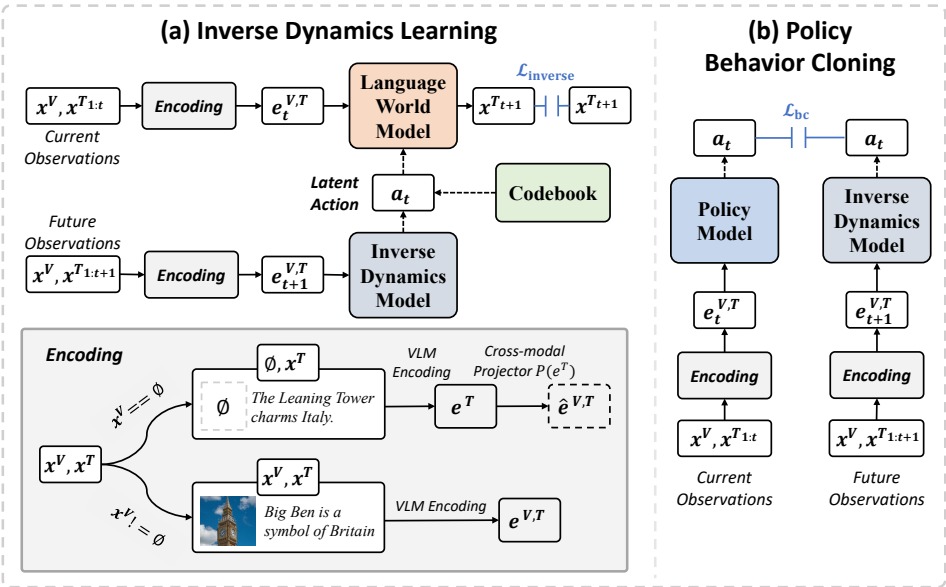


Figure 2: Pipeline for constructing the latent action space. (a) **Inverse dynamics learning**: Given future observations, the inverse dynamics model infers a discrete latent action from a learnable codebook; the language world model then uses this latent action and current observations to reconstruct the next token $x^{T_{t+1}}$. The language world model, inverse dynamics model, and codebook are jointly trained. (b) **Policy behavior cloning**: A policy model is trained to predict the same latent actions as those inferred by the inverse dynamics model, using only current observations.

Similarly, P' is trained on $\mathcal{D}^{VT}$ using the symmetric loss $\mathcal{L}_{\text{vt2t}}$, defined as:

$$\mathcal{L}_{\text{vt2t}} = \mathbb{E}_{\mathcal{D}^{VT}}\left[\sum_{t=1}^{m-1}\frac{1}{2}\left(\left\|\frac{e_t^T-\nu_t}{\tau_t}\right\|^2 + \|\log\tau_t^2\|_1\right)\right], \tag{4}$$

where the expectation is taken over sequences $(x^V, x^{T_{1:m}}) \sim \mathcal{D}^{VT}$, $e_t^T = f_{\text{VLM}}(x^{T_{1:t}})$ denotes the text embedding, and $(\nu_t, \tau_t) = P'(e_t^{V,T} = f_{\text{VLM}}(x^V, x^{T_{1:t}}))$. The total loss for *Step 1* is:

$$\mathcal{L}_{\text{proj}_1} = \mathcal{L}_{\text{t2vt}} + \mathcal{L}_{\text{vt2t}}. \tag{5}$$

Step 2: Jointly training on paired image-text data and text-only data We now jointly train P and P' on paired data $\mathcal{D}^{VT}$ and text-only data $\mathcal{D}^T$. The total objective is:

$$\mathcal{L}_{\text{proj}_2} = \mathcal{L}_{\text{t2vt}} + \mathcal{L}_{\text{vt2t}} + \mathcal{L}_{\text{cycle}} \tag{6}$$

where $\mathcal{L}_{\text{t2vt}}$ (Eq. 3) and $\mathcal{L}_{\text{vt2t}}$ (Eq. 4) are computed over $\mathcal{D}^{VT}$, and $\mathcal{L}_{\text{cycle}}$ denotes a novel cycle consistency loss computed on text-only data $\mathcal{D}^T$.

The cycle consistency loss $\mathcal{L}_{\text{cycle}}$ is defined as:

$$\mathcal{L}_{\text{cycle}} = \mathbb{E}_{\mathcal{D}^{T}}\left[\sum_{t=1}^{m-1}\frac{1}{2}\left(\left\|\frac{e_t^T-\nu_t}{\tau_t}\right\|^2 + \|\log\tau_t^2\|_1\right)\right], \tag{7}$$

where the expectation is taken over text-only sequences $x^{T_{1:m}} \sim \mathcal{D}^T$, $e_t^T = f_{\text{VLM}}(x^{T_{1:t}})$, and $(\mu_t, \sigma_t) = P(e_t^T)$, and $(\nu_t, \tau_t) = P'(\mu_t)$.

3.2.2 Policy Behavior Cloning

During RL exploration and inference, future observations are unavailable, making the inverse dynamics model f_{inverse} inapplicable. Thus, we train a policy model π_θ via behavior cloning to mimic latent actions inferred by f_{inverse} (Fig. 2 (b)). Specifically, for samples from the mixed corpus $\mathcal{D}^{\text{mix}} = \mathcal{D}^{VT} \cup \mathcal{D}^T$, we compute the loss as:

$$\mathcal{L}_{\text{bc}} = \mathbb{E}_{\mathcal{D}^{\text{mix}}}\left[-\sum_{t=1}^{m-1}\log\pi_\theta\left(a_t^* = f_{\text{inverse}}(e_{t+1}^{V,T}) \mid e_t^{V,T}\right)\right], \tag{8}$$

where the expectation is taken over sequences $(x^V, x^{T_{1:m}}) \sim \mathcal{D}^{\text{mix}}$, with $e_t^{V,T}$ defined as in Eq. 2.

3.3 Latent Action Reinforcement Learning

On downstream multimodal conversational tasks, we perform reinforcement learning at the policy model level, as illustrated in Fig. 3. For each prompt $(x^V, x^{T_{1:p}}) \sim \mathcal{D}_{\text{rl}}$ with the prompt length p, the policy π_θ and the world model f_{world} jointly generate response $x^{T_{p+1:m}}$ auto-regressively, i.e., at each step $t = p, \ldots, m-1$, $a_t \sim \pi_\theta(\cdot|x^V, x^{T_{1:t}}), x^{T_{t+1}} = f_{\text{world}}(x^V, x^{T_{1:t}}, a_t)$, with maximum length m. We optimize π_θ by maximizing the expected rewards:

$$\mathcal{J}(\theta) = \mathbb{E}_{(x^V, x^{T_{1:p}})\sim\mathcal{D}_{\text{rl}}}\left[R(x^{T_{p+1:m}})\right], \tag{9}$$

where $R(\cdot)$ denotes the reward function.

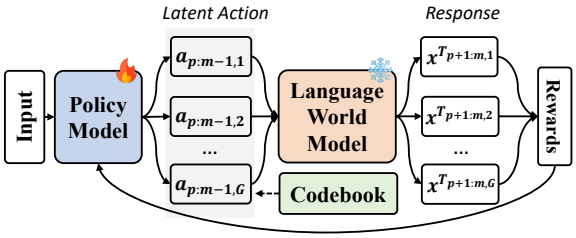


Figure 3: Illustrations of latent action RL. The language world model is frozen, while the policy model is optimized to select latent actions from the codebook that steer the generated responses toward higher rewards.

Algorithm 1 Latent Action Space Learning and Latent Action RL

Stage 1: Latent Action Space Learning

Initialize $f_{\text{world}}, f_{\text{inverse}}, \mathcal{C}$ by minimizing $\mathcal{L}_{\text{inverse}}$ (Eq. 1) on $\mathcal{D}^{VT}$.
Initialize the cross-modal projectors P, P' by minimizing $\mathcal{L}_{\text{proj}_1}$ (Eq. 5) on $\mathcal{D}^{VT}$.
Jointly optimize $f_{\text{world}}, f_{\text{inverse}}, \mathcal{C}, P, P'$ by minimizing $\mathcal{L}_{\text{inverse}}$ (Eq. 1) and $\mathcal{L}_{\text{proj}_2}$ (Eq. 6) on $\mathcal{D}^{VT} \cup \mathcal{D}^{T}$.
Initialize the policy model π_θ by minimizing $\mathcal{L}_{\text{bc}}$ (Eq. 8) on $\mathcal{D}^{VT} \cup \mathcal{D}^{T}$.

Stage 2: Latent Action RL

Sample $(x^V, x^{T_{1:p}}) \sim \mathcal{D}_{\text{rl}}$:
  Roll out $x^{T_{p+1:m}}$ via $a_t \sim \pi_\theta(\cdot|x^V, x^{T_{1:t}})$, $x^{T_{t+1}} = f_{\text{world}}(x^V, x^{T_{1:t}}, a_t), t = p, ..., m-1$.
  Compute reward $R(x^{T_{p+1:m}})$.
  Optimize π_θ by maximizing $\mathcal{J}(\theta)$ (Eq. 9).

We summarize our framework in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Models We build the language world model, inverse dynamics model, and policy model upon the same foundation vision-language model. Specifically, we use `Qwen2.5-VL-3B-Instruct` and `Qwen2.5-VL-7B-Instruct` (Bai et al., 2025b) for main experiments. The latent action space is implemented as a codebook with size $|\mathcal{C}| = 128$.

Datasets During the latent action space construction stage (Section 3.2), we use a mixture of paired image-text corpora $\mathcal{D}^{VT}$ and text-only corpora $\mathcal{D}^{T}$. For $\mathcal{D}^{VT}$, we collect image-caption pairs from `Conceptual-12M` (Changpinyo et al., 2021), multimodal news articles from `N24News` (Wang et al., 2022), and multimodal Wikipedia data from `WikiWeb2M` (Burns et al., 2023), totaling 14 million images and 1 billion text tokens. For $\mathcal{D}^{T}$, we collect text-only data mainly from the `SlimPajama-627B` dataset (Soboleva et al., 2023), which contains 627 billion text tokens.

For latent action RL (Section 3.3), we evaluate our method on two downstream tasks: 1) multimodal role-playing conversation on `MMRole` (Dai et al., 2025), where we focus on the challenging `Comment` subset; we train on the in-distribution (ID) split and evaluate on ID and out-of-distribution (OOD) test sets; 2) multimodal personalized conversation on `PCogAlignBench` (Li et al., 2025a), where we train the agent on the `LS1` set and evaluate on `LS1` and `LS2` test sets.

Evaluation Metrics We adopt the *LLM-as-a-Judge* metric to evaluate model performance, using prompt templates validated by Dai et al. (2025); Li et al. (2025a), which show high correlation with human judgments. For each sample, the LLM judge scores both the model and ground-truth responses across benchmark-specific dimensions, with scores ranging 1-10. Then, following (Dai et al., 2025), we report the ratio of the model's average score to the ground-truth response's average score across all evaluation dimensions. We report the mean and standard deviation across three evaluation runs.

Baselines We consider two categories of baselines: 1) Non-RL baselines: the naive **Prompt** and supervised fine-tuning (**SFT**); 2) RL-based methods, where we compare two optimization strategies, token-level and latent action RL, using four algorithms: a) Group Relative Policy Optimization (**GRPO**) (Shao et al., 2024), b) **Dr. GRPO** (Liu et al., 2025), c) Decoupled Clip and Dynamic Sampling Policy Optimization (**DAPO**) (Yu et al., 2025), and d) Beta Normalization Policy Optimization (**BNPO**) (Xiao et al., 2025). The reward functions are kept the same for methods. Please refer to the Appendix B for more experimental details.

4.2 Main Results

Overall Performance Table 1 reports the experimental results of token-level baselines and our proposed latent action level RL. Based on these results, we have made the following observations. 1) Our method achieves superior performance across diverse tasks and datasets. On average, it outperforms token-level RL by 4% (averaged over all settings). 2) Our latent action framework is RL-agnostic and readily compatible with diverse policy optimization algorithms, including GRPO, Dr. GRPO, DAPO,

	Method	MMRole		PCogAlignBench		Average
		ID	*OOD*	*LS1*	*LS2*	
Qwen2.5-VL-3B-Instruct	Prompt	$0.728_{\pm 0.005}$	$0.687_{\pm 0.025}$	$0.678_{\pm 0.003}$	$0.676_{\pm 0.002}$	$0.692_{\pm 0.009}$
	SFT	$0.843_{\pm 0.002}$	$0.809_{\pm 0.012}$	$0.808_{\pm 0.009}$	$0.810_{\pm 0.005}$	$0.817_{\pm 0.007}$
	GRPO (Token)	$0.838_{\pm 0.017}$	$0.796_{\pm 0.027}$	$0.845_{\pm 0.007}$	$\mathbf{0.845}_{\pm 0.004}$	$0.831_{\pm 0.014}$
	GRPO (Latent Action)	$\mathbf{0.949}_{\pm 0.007}$	$\mathbf{0.915}_{\pm 0.065}$	$\mathbf{0.871}_{\pm 0.011}$	$0.837_{\pm 0.010}$	$\mathbf{0.893}_{\pm 0.023}$
	Dr.GRPO (Token)	$0.867_{\pm 0.011}$	$0.823_{\pm 0.002}$	$0.835_{\pm 0.008}$	$0.834_{\pm 0.012}$	$0.840_{\pm 0.008}$
	Dr.GRPO (Latent Action)	$\mathbf{0.953}_{\pm 0.016}$	$\mathbf{0.916}_{\pm 0.038}$	$\mathbf{0.874}_{\pm 0.009}$	$\mathbf{0.840}_{\pm 0.009}$	$\mathbf{0.896}_{\pm 0.018}$
	DAPO (Token)	$0.856_{\pm 0.003}$	$0.805_{\pm 0.033}$	$0.835_{\pm 0.008}$	$0.828_{\pm 0.008}$	$0.831_{\pm 0.013}$
	DAPO (Latent Action)	$\mathbf{0.941}_{\pm 0.016}$	$\mathbf{0.889}_{\pm 0.009}$	$\mathbf{0.879}_{\pm 0.011}$	$\mathbf{0.835}_{\pm 0.006}$	$\mathbf{0.886}_{\pm 0.010}$
	BNPO (Token)	$0.860_{\pm 0.012}$	$0.801_{\pm 0.038}$	$0.849_{\pm 0.008}$	$0.836_{\pm 0.007}$	$0.836_{\pm 0.016}$
	BNPO (Latent Action)	$\mathbf{0.940}_{\pm 0.004}$	$\mathbf{0.901}_{\pm 0.014}$	$\mathbf{0.872}_{\pm 0.007}$	$\mathbf{0.836}_{\pm 0.008}$	$\mathbf{0.887}_{\pm 0.008}$
Qwen2.5-VL-7B-Instruct	Prompt	$0.839_{\pm 0.006}$	$0.821_{\pm 0.024}$	$0.721_{\pm 0.003}$	$0.710_{\pm 0.003}$	$0.773_{\pm 0.009}$
	SFT	$0.885_{\pm 0.003}$	$0.856_{\pm 0.013}$	$0.808_{\pm 0.005}$	$0.799_{\pm 0.004}$	$0.837_{\pm 0.006}$
	GRPO (Token)	$0.892_{\pm 0.004}$	$0.840_{\pm 0.014}$	$0.870_{\pm 0.016}$	$0.851_{\pm 0.012}$	$0.863_{\pm 0.011}$
	GRPO (Latent Action)	$\mathbf{0.920}_{\pm 0.005}$	$\mathbf{0.872}_{\pm 0.016}$	$\mathbf{0.898}_{\pm 0.009}$	$\mathbf{0.852}_{\pm 0.010}$	$\mathbf{0.885}_{\pm 0.010}$
	Dr.GRPO (Token)	$0.892_{\pm 0.006}$	$0.854_{\pm 0.009}$	$0.854_{\pm 0.006}$	$0.839_{\pm 0.004}$	$0.860_{\pm 0.006}$
	Dr.GRPO (Latent Action)	$\mathbf{0.916}_{\pm 0.010}$	$\mathbf{0.864}_{\pm 0.020}$	$\mathbf{0.897}_{\pm 0.008}$	$\mathbf{0.851}_{\pm 0.015}$	$\mathbf{0.882}_{\pm 0.013}$
	DAPO (Token)	$0.892_{\pm 0.004}$	$0.842_{\pm 0.025}$	$0.844_{\pm 0.013}$	$0.828_{\pm 0.007}$	$0.852_{\pm 0.012}$
	DAPO (Latent Action)	$\mathbf{0.920}_{\pm 0.009}$	$\mathbf{0.863}_{\pm 0.017}$	$\mathbf{0.903}_{\pm 0.012}$	$\mathbf{0.850}_{\pm 0.005}$	$\mathbf{0.884}_{\pm 0.011}$
	BNPO (Token)	$0.894_{\pm 0.004}$	$\mathbf{0.859}_{\pm 0.029}$	$0.850_{\pm 0.007}$	$0.836_{\pm 0.004}$	$0.860_{\pm 0.011}$
	BNPO (Latent Action)	$\mathbf{0.916}_{\pm 0.006}$	$0.842_{\pm 0.018}$	$\mathbf{0.901}_{\pm 0.009}$	$\mathbf{0.852}_{\pm 0.012}$	$\mathbf{0.878}_{\pm 0.011}$

Table 1: Performance comparison on MMRole and PCogAlignBench, using the *LLM-as-a-Judge* metric. Results are averaged over three runs. We conduct experiments using various VLMs, including Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct. Best results are in **bold** on each RL algorithm.

and BNPO, yielding consistent gains over baselines. 3) The improvements brought by latent actions are consistently observed in both 3B and 7B models, demonstrating the scalability of our approach.

Performance on Fine-grained Dimensions To thoroughly evaluate the performance of multimodal conversational agents trained with latent actions across various fine-grained conversational dimensions, following prior work (Dai et al., 2025; Li et al., 2025a), we assess eight dimensions on `MMRole`: 1) Instruction Adherence (IA), 2) Fluency (Flu), 3) Coherency (Coh), 4) Image-Text Relevance (ITR), 5) Response Accuracy (RA), 6) Personality Consistency (OC), 7) Knowledge Consistency (KC), and 8) Tone Consistency (TC). On `PCogAlignBench`, we evaluate: 1) Role-Set Awareness (RSA), 2) Body Behavior Awareness (BBA), 3) Mind Feelings Awareness (MFA), 4) Contextual Awareness (CA), and 5) Conversational Flow (CF). We present the comparison results in Fig. 4, with detailed results provided in Appendix C.2.

As shown in Figure 4, we make the following observations: 1) Overall, our methods outperform token-level baselines across all evaluated dimensions. 2) While both our method and the baselines achieve strong performance on basic conversational capabilities, such as Fluency (Flu) and Conversational Flow (CF), our approach demonstrates substantially more pronounced improvements on more challenging personalized dimensions, such as Tone Consistency (TC) on `MMRole`.

4.3 Ablation Study

To assess the contribution of main components in our method, we conduct ablation study using three variants. 1) *Ours w/o cycle consistency*: We remove the cycle consistency loss during cross-modal projector training, and instead directly apply the projector trained only on paired image-text data, i.e., removing $\mathcal{L}_{\text{cycle}}$ in Eq. 6; 2) *Ours w/o cross-modal projector*: We remove the cross-modal projector entirely, and learn the latent action codebook directly from text-only representations e^T; 3) *Ours*

Method	MMRole		PCogAlignBench		Avg.
	ID	*OOD*	*LS1*	*LS2*	
Ours	**0.949**$_{\pm 0.007}$	**0.915**$_{\pm 0.065}$	**0.871**$_{\pm 0.011}$	**0.837**$_{\pm 0.010}$	**0.893**$_{\pm 0.023}$
Ours *w/o cycle consistency*	0.921$_{\pm 0.005}$	0.878$_{\pm 0.023}$	0.858$_{\pm 0.007}$	0.825$_{\pm 0.013}$	0.870$_{\pm 0.012}$
Ours *w/o cross-modal projector*	0.944$_{\pm 0.014}$	0.901$_{\pm 0.014}$	0.858$_{\pm 0.010}$	0.819$_{\pm 0.013}$	0.880$_{\pm 0.013}$
Ours *w/o text-only data*	0.932$_{\pm 0.010}$	0.861$_{\pm 0.036}$	0.851$_{\pm 0.007}$	0.817$_{\pm 0.006}$	0.865$_{\pm 0.015}$

Table 2: Ablation study on main components of our method. We evaluate on MMRole and PCogAlignBench using the *LLM-as-a-Judge* metric. Results are averaged over three runs. All variants are fine-tuned with GRPO based on Qwen2.5-VL-3B-Instruct. Best results are in **bold**.

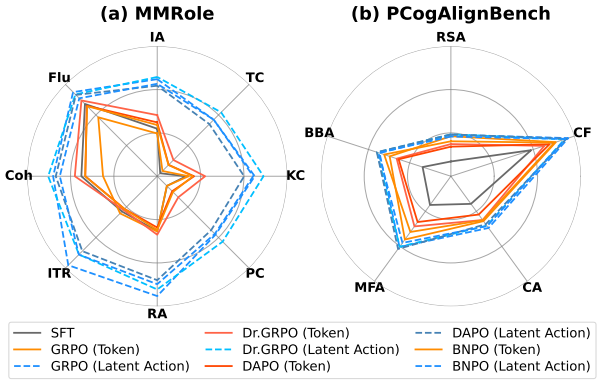


Figure 4: Fine-grained performance comparison on (a) MMRole and (b) PCogAlignBench. Results using latent actions are shown with dashed lines, while results using token-level RL are plotted with solid lines.

w/o text-only data: We construct the latent action space using only the limited paired multimodal corpus, excluding all text-only data. The results of ablation study are shown in Table 2.

From Table 2, we can make the following observations. 1) Removing the cycle consistency loss leads to an average performance drop of 2.3%, indicating that fine-tuning the projector on large-scale text-only data via cycle consistency loss is crucial for improving its robustness. 2) Eliminating the cross-modal projector causes a noticeable decline in performance. This suggests that directly learning the latent action space from text-only embeddings may introduce a unimodal bias, i.e., the trained latent action policy model overly relies textual representations and fail to effectively handle multimodal scenarios. 3) Solely leveraging paired multimodal data results in the largest performance degradation, particularly in out-of-distribution settings (e.g., `OOD` on `MMRole` and `LS2` on `PCogAlignBench`). This highlights that the limited diversity and coverage of paired image-text corpora constrain the generalization capability of latent action policy models.

4.4 Analysis

Rollout Diversity with Latent Actions Benefiting from the reduced action space, the constructed latent action space is expected to improve the agent's rollout diversity during RL exploration, i.e., generating more diverse responses. Prior work has shown that such diversity is critical for improving the upper bound of RL performance (Li et al., 2025b; Yu et al., 2025).

Following Jia et al. (2025), we quantify rollout diversity via *semantic diversity*, as it reflects both linguistic diversity and response quality. Concretely, as shown in Fig. 3, for each prompt $(x^T, x^{T_{1:p}})$ in the RL training set $\mathcal{D}_{\text{RL}}$, the agent generates G responses $\{x^{T_{p+1:m,i}}\}_{i=1}^{G}$, with p as the prompt length and m as the maximum length. We calculate the semantic diversity as:

$$\frac{G(G-1)}{\sum_{i=1}^{G}\sum_{j=1,j\neq i}^{G}\text{Sim}(x^{T_{p+1:m,i}}, x^{T_{p+1:m,j}})}, \tag{10}$$

where $\text{Sim}(\cdot,\cdot)$ denotes the embedding similarity between two responses and we adopt BGE-M3 (Chen et al., 2024) as the embedding model. We report the average semantic diversity over all samples in the RL training set.

In Table 3, we compare the rollout diversity of token based and latent action based RL algorithms. From Table 3, we observe that latent action RL consistently and significantly outperforms token-level RL in rollout diversity, demonstrating the superior exploration efficiency. We also provide a case study in Appendix C.3 to illustrate the improvements in rollout diversity intuitively.

Computational Budget To assess the computational overhead introduced by our latent action framework, we analyze the time cost during RL training. Specifically, we consider the time cost in two stages: 1) *Rollout*: generating multiple candidate responses per prompt; 2) *Policy update*: updat-

Method	MMRole	PCogAlignBench
GRPO (Token)	$1.079_{\pm 0.230}$	$1.043_{\pm 0.197}$
GRPO (Latent Action)	$\mathbf{1.256}_{\pm 0.302}$	$\mathbf{1.200}_{\pm 0.326}$
Dr.GRPO (Token)	$1.074_{\pm 0.224}$	$1.259_{\pm 0.249}$
Dr.GRPO (Latent Action)	$\mathbf{1.252}_{\pm 0.297}$	$\mathbf{1.323}_{\pm 0.278}$
DAPO (Token)	$1.075_{\pm 0.230}$	$1.040_{\pm 0.182}$
DAPO (Latent Action)	$\mathbf{1.254}_{\pm 0.302}$	$\mathbf{1.131}_{\pm 0.253}$
BNPO (Token)	$1.078_{\pm 0.224}$	$1.261_{\pm 0.257}$
BNPO (Latent Action)	$\mathbf{1.297}_{\pm 0.303}$	$\mathbf{1.321}_{\pm 0.286}$

Table 3: Rollout diversity during RL exploration. Higher values indicate better rollout diversity. Best results are in **bold**.

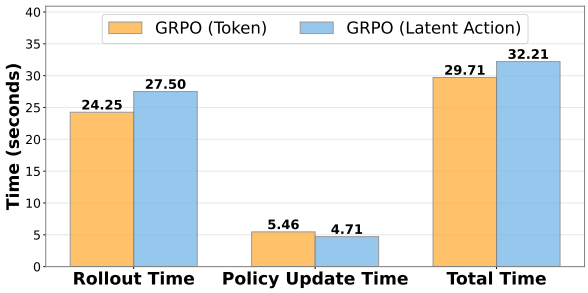


Figure 5: Time cost per step during RL training, including rollout, policy update, and total time.

ing the policy model using the computed rewards. We present the time cost per RL step of our method and the baseline in Fig. 5, using GRPO as an example with a rollout batch size of 8.

As illustrated in Fig. 5, our latent action based method incurs a 1.13× slowdown in rollout time, due to the additional latent action prediction step. However, policy updates in latent action RL require only 0.86× the time of the baseline, as the optimization involves adjusting the policy's output distribution over a compact latent action space, rather than the full token vocabulary. Overall, the total RL training time is only 1.08× that of token-level RL.

5 Related Work

Multimodal Conversational Agents Recent advances in vision-language models (VLMs) (Bai et al., 2025b) have enabled increasingly capable multimodal conversational agents (MCAs) (Yao et al., 2025), such as multimodal role-playing agents (Dai et al., 2025) and personalized assistants (Nguyen et al., 2024; Li et al., 2025a), which hold significant promise in fields like entertainment (Mehta et al., 2022) and personalized education (Griol et al., 2014). Initial efforts to build MCAs primarily rely on supervised fine-tuning (Li et al., 2024), but often suffer from poor generalization. Recently, RL has been widely explored for fine-tuning MCAs and has demonstrated strong generalization performance (Zhou et al., 2025; Chu et al., 2025). However, fine-tuning MCAs via RL faces challenges in handling the extremely large text token space. To address this, we propose constructing a compact latent action space for RL fine-tuning, which enables efficient policy learning.

Reinforcement Learning with Latent Actions In many real-world scenarios, only observation-only data are available, such as expert demonstration videos of robots where explicit action labels are missing (Torabi et al., 2019). To address this challenge, prior works leverage the learning from observation mechanism (Seo et al., 2022; Baker et al., 2022) to infer latent actions from observation-only data, which are then used for RL fine-tuning of agents. For instance, Zhang et al. (2024b); Gao et al. (2025) learn latent actions from videos to control video generation, while Ye et al. (2025); Bu et al. (2025) extract latent actions from robot manipulation videos and use them for robot policy learning. These constructed latent actions not only enhance controllability (Bruce et al., 2024) but also enable better transferability across different tasks due to their higher-level nature (Jang et al., 2025).

The most relevant work to ours is CoLA (Jia et al., 2025), which introduces latent actions into RL fine-tuning of LLMs. However, when constructing the latent action space for multimodal conversational agents, the scarcity of paired image-text data hinders learning a latent space with sufficient coverage. To overcome this, we leverage both paired image-text data and massive text-only data to construct the latent space, using a cross-modal projector trained with a novel cycle-consistency loss.

6 Conclusion

In this work, we propose to learn a compact latent action space for reinforcement learning (RL) fine-tuning of multimodal conversational agents (MCAs). To construct this latent space, we leverage both paired image-text data and abundant text-only data, using a cross-modal projector trained with a novel cycle-consistency loss, which improves the coverage of latent actions while avoiding potentially unimodal bias. We evaluate our approach on two tasks, including multimodal role-playing and multimodal personalized conversation, and demonstrate significant improvements over competitive baselines across various RL algorithms.

Limitations

We acknowledge the following limitations in our work. First, the additional latent action prediction step increases RL training time by 1.08× and inference latency by 1.13×. Second, due to constraints of computational resources, we evaluate our latent action based approach on multimodal conversational tasks, and leave validation on more tasks to future work, such as visual mathematical reasoning.

Ethics Considerations

Our work is entirely at the methodological level, which means that there will not be any negative social impacts.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025b. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. 2024. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*.

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

Andrea Burns, Krishna Srinivasan, Joshua Ainslie, Geoff Brown, Bryan A. Plummer, Kate Saenko, Jianmo Ni, and Mandy Guo. 2023. A suite of generative tasks for multi-level multimodal webpage understanding. In *The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*.

Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T Ash, Akshay Krishnamurthy, and Dylan J Foster. 2025. The coverage principle: How pre-training enables post-training. *arXiv preprint arXiv:2510.15020*.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand. Association for Computational Linguistics.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*.

Yanqi Dai, Huanran Hu, Lei Wang, Shengjie Jin, Xu Chen, and Zhiwu Lu. 2025. Mmrole: A comprehensive framework for developing and evaluating multimodal role-playing agents. In *The Thirteenth International Conference on Learning Representations*.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. 2025. Adaworld: Learning adaptable world models with latent actions. In *Forty-second International Conference on Machine Learning*.

David Griol, José Manuel Molina, and Araceli Sanchís De Miguel. 2014. Developing multimodal conversational agents for an enhanced e-learning experience. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 3(1):13–26.

Sharut Gupta, Shobhita Sundaram, Chenyu Wang, Stefanie Jegelka, and Phillip Isola. 2025. Better together: Leveraging unpaired multimodal data for stronger unimodal models. *arXiv preprint arXiv:2510.08492*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer

Huang, Kaushil Kundalia, Yen-Chen Lin, et al. 2025. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*.

Chengxing Jia, Ziniu Li, Pengyuan Wang, Yi-Chen Li, Zhenyu Hou, Yuxiao Dong, and Yang Yu. 2025. Controlling large language model with latent action. In *Forty-second International Conference on Machine Learning*.

Zhengyao Jiang, Tianjun Zhang, Michael Janner, Yueying Li, Tim Rocktäschel, Edward Grefenstette, and Yuandong Tian. 2023. Efficient planning in a compact latent action space. In *The Eleventh International Conference on Learning Representations*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*.

Yongqi Li, Shen Zhou, Xiaohu Li, Xin Miao, Jintao Wen, Mayi Xu, Jianhao Chen, Birong Pan, Hankun Kang, Yuanyuan Zhu, Ming Zhong, and Tieyun Qian. 2025a. Aligning VLM assistants with personalized situated cognition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9813–9839, Vienna, Austria. Association for Computational Linguistics.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025b. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective.

Aditya Mehta, Yug Kunjadiya, Aniket Kulkarni, and Manav Nagar. 2022. Exploring the viability of conversational ai for non-playable characters: A comprehensive survey. In *2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST)*, pages 96–102. IEEE.

Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. 2024. Yo'llava: Your personalized language and vision assistant. *Advances in Neural Information Processing Systems*, 37:40913–40951.

Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. 2021. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. 2022. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pages 19561–19579. PMLR.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Recent advances in imitation learning from observation. *ArXiv*, abs/1905.13566.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Zhen Wang, Xu Shan, Xiangxie Zhang, and Jie Yang. 2022. N24news: A new dataset for multimodal news classification. In *Proceedings of the Language Resources and Evaluation Conference*, pages 6768–6775, Marseille, France. European Language Resources Association.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. 2025. Helpsteer3-preference: Open human-annotated preference data across diverse tasks and languages.

Changyi Xiao, Mengdi Zhang, and Yixin Cao. 2025. Bnpo: Beta normalization policy optimization.

Huanjin Yao, Ruifei Zhang, Jiaxing Huang, Jingyi Zhang, Yibo Wang, Bo Fang, Ruolin Zhu, Yongcheng Jing, Shunyu Liu, Guanbin Li, et al. 2025. A survey on agentic multimodal large language models. *arXiv preprint arXiv:2510.10991*.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars

Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. 2025. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yedi Zhang, Peter E Latham, and Andrew M Saxe. 2024a. Understanding unimodal bias in multimodal deep linear networks. In *International Conference on Machine Learning*, pages 59100–59125. PMLR.

Zhilong Zhang, Ruifeng Chen, Junyin Ye, Yihao Sun, Pengyuan Wang, Jingcheng Pang, Kaiyuan Li, Tianshuo Liu, Haoxin Lin, Yang Yu, et al. 2024b. Whale: Towards generalizable and scalable world models for embodied decision-making. *arXiv preprint arXiv:2411.05619*.

Guanghao Zhou, Panjia Qiu, Cen Chen, Jie Wang, Zheming Yang, Jian Xu, and Minghui Qiu. 2025. Reinforced mllm: A survey on rl-based reasoning in multimodal large language models. *arXiv preprint arXiv:2504.21277*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

A Details on Model Design

A.1 Language World Model

The language world model $f_{\text{world}}(x^{T_{t+1}} \mid x^V, x^{T_{1:t}}, a_t)$ predicts the next token $x^{T_{t+1}}$ autoregressively given the current multimodal context $(x^V, x^{T_{1:t}})$ and a latent action a_t predicted by the inverse dynamics model (during the latent action space learning) or the policy model (during latent action RL and inference). It consists of two core modules, reusing some components from the original VLM:

Encode Module This module encodes the input $(x^V, x^{T_{1:t}})$ into a context embedding $e_t^{V,T} \in \mathbb{R}^d$, using the transformer blocks of the original VLM.

Merge Module This module fuses the context embedding $e_t^{V,T}$ and the latent action embedding $c_{a_t} \in \mathbb{R}^d$ (where c_{a_t} is the code vector in $\mathcal{C}$ corresponding to the latent action a_t) to produce the next-token prediction. Specifically, a two-layer MLP $f_{\text{mlp}} : \mathbb{R}^{2d} \to \mathbb{R}^d$ takes the concatenation $[e_t^{V,T}; c_{a_t}]$ as input and outputs a merged representation $e_t^{\text{mlp}} = f_{\text{mlp}}([e_t^{V,T}; c_{a_t}])$. Then, the merged vector e_t^{mlp} is fed into the original VLM's language modeling head f_{head}, yielding the token prediction distribution $p(x^{T_{t+1}} \mid \cdot) = f_{\text{head}}(e_t^{\text{mlp}})$. The next token $x^{T_{t+1}}$ is selected from this distribution.

A.2 Inverse Dynamics Model

The inverse dynamics model $f_{\text{inverse}}(a_t | x^V, x^{T_{1:t+1}}))$ is designed to take future observations $(x^V, x^{T_{1:t+1}})$ as input, and extracts the latent action a_t for the current step t. It consists of three core modules.

Encode Module The input $(x^V, x^{T_{1:t+1}})$ is encoded into $e_{t+1}^{V,T} \in \mathbb{R}^d$ using the transformer blocks of the original VLM. When $x^V = \emptyset$ (text-only sequences), the text embedding $e_{t+1}^T = f_{\text{VLM}}(x^{T_{1:t+1}})$ is projected to the image-text embedding via the cross-modal projector P, i.e., $\hat{e}_{t+1}^{V,T} = P(e_{t+1}^T)$, as illustrated in Fig. 2.

Inverse Transformer Layers To adapt the VLM embedding to the latent action space, the obtained embedding $e_{t+1}^{V,T}$ is processed by 4-layer Transformer blocks, yielding a representation $\tilde{e}_{t+1}^{V,T} \in \mathbb{R}^d$.

Inverse Action Head Following Jia et al. (2025), we adopt a *direct code assignment* strategy to avoid code collapse. Specifically, a linear head (inverse action head) maps $\tilde{e}_{t+1}^{V,T}$ to logits $\mathbf{l}_t \in \mathbb{R}^{|\mathcal{C}|}$ over the codebook indices. During inverse dynamics learning, we apply the Gumbel-Softmax and a reparameterization trick to obtain a differentiable soft assignment:

$$\mathbf{g}_t = \text{GumbelSoftmax}(\mathbf{l}_t), \quad \hat{\mathbf{o}}_t = (\mathbf{o}_t - \mathbf{g}_t)_{\text{sg}} + \mathbf{g}_t,$$

where $\mathbf{o}_t$ is the hard one-hot vector ($\arg\max$ of $\mathbf{l}_t$), and $(\cdot)_{\text{sg}}$ denotes stop-gradient. The final latent action embedding is $c_{a_t} = \hat{\mathbf{o}}_t^\top \mathcal{C}$, which is then used by the language world model.

A.3 Policy Model

The policy $\pi_\theta(a_t \mid x^V, x^{T_{1:t}})$ predicts the latent action a_t from the current context $(x^V, x^{T_{1:t}})$. Its architecture mirrors f_{inverse}, which includes: 1) the encode module, 2) policy transformer layers (8-layer), and 3) policy action head.

A.4 Codebook for the Latent Action Space

The latent action space is defined by a codebook $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^d$ with $K = 128$. Each code vector c_k is initialized independently via Kaiming uniform initialization (He et al., 2015). Given a latent action index $a_t \in \{1, \ldots, K\}$, the corresponding latent action embedding is retrieved as $c_{a_t} \in \mathcal{C}$.

A.5 Cross-modal Projector

The cross-modal projector P is implemented as a dual-MLP module: given a text embedding e_t^T, the first MLP outputs the mean vector μ_t, and the second MLP outputs the log standard deviation vector $\log \sigma_t$ (for numerical stability), forming a diagonal Gaussian distribution $\mathcal{N}(\mu_t, \text{diag}(\sigma_t^2))$ in the image-text embedding space.

B Experimental Details

B.1 Details on Datasets

Corpora for Constructing the Latent Action Space To construct the latent action space in an unsupervised manner, we collect large-scale paired image-text and text-only corpora. For paired image-text data, we use: (1) image-caption pairs from `Conceptual-12M` (Changpinyo et al., 2021); (2) multimodal news articles from `N24News` (Wang et al., 2022); and (3) multimodal Wikipedia articles from `WikiWeb2M` (Burns et al., 2023), comprising 14M images and 1B text tokens in total. For text-only data, we primarily sample 500K sequences from `SlimPajama-627B` (Soboleva et al.,

2023) due to computational constraints, and additionally include 40K alignment corpora from `HelpSteer3` (Wang et al., 2025) to preserve the original VLM's safety and preference alignment during latent space learning. To ensure fair comparison, we analyze data exposure in Appendix C.1 and find that downstream task performance does not benefit from the above corpora, confirming that observed improvements stem from methodological advances.

B.2 Details on Evaluation Metric

We adopt *LLM-as-a-Judge* metrics to evaluate model performance, using prompt templates validated by Dai et al. (2025); Li et al. (2025a), which show high correlation with human judgments. The evaluation prompt templates used on `MMRole` and `PCogAlignBench` are shown in Table 4. We adopt the `Qwen3-235B-A22B` by the Qwen3 API platform as the judge model.

B.3 Training Details

Baseline Methods For the SFT baseline, we fine-tune the VLM with a learning rate of 5×10^{-6} for 2 epochs. For token-level RL baselines, we use a rollout size of 8, a per-step batch size of 32, and train for 100 RL steps with a constant learning rate of 1×10^{-6}. For all RL methods, we use 50% of the training data to initialize the model via SFT, followed by RL fine-tuning on the remaining 50%. During RL rollouts, we set the sampling temperature to 1.0 for all methods.

Latent Action Space Learning As outlined in Algorithm 1, the latent action space learning procedure consists of the following four stages:

1. Initialize $f_{\text{world}}, f_{\text{inverse}}, \mathcal{C}$ by minimizing $\mathcal{L}_{\text{inverse}}$ (Eq. 1) on $\mathcal{D}^{VT}$. *Training details*: learning rate = 1×10^{-4}, cosine decay with minimum learning rate 1×10^{-5}, batch size = 16, max sequence length = 2048, 1 epoch.

2. Initialize the cross-modal projectors P, P' by minimizing $\mathcal{L}_{\text{proj}_1}$ (Eq. 5) on $\mathcal{D}^{VT}$. *Training details*: learning rate = 1×10^{-3}, cosine decay, batch size = 16, 1 epoch.

3. Jointly optimize $f_{\text{world}}, f_{\text{inverse}}, \mathcal{C}, P, P'$ by minimizing $\mathcal{L}_{\text{inverse}}$ (Eq. 1) and $\mathcal{L}_{\text{proj}_2}$ (Eq. 6) on $\mathcal{D}^{VT} \cup \mathcal{D}^{T}$. *Training details*: learning rate = 1×10^{-4}, cosine decay with minimum learning rate 1×10^{-5}, batch size = 16, max sequence length = 2048, 1 epoch.

4. Initialize the policy model π_θ by minimizing $\mathcal{L}_{\text{bc}}$ (Eq. 8) on $\mathcal{D}^{VT} \cup \mathcal{D}^{T}$. *Training details*: learning rate = 1×10^{-4}, cosine decay, batch size = 16, max sequence length = 2048, 1 epoch.

Latent Action RL We adopt the same RL hyperparameters as the token-level baselines: rollout size of 8, per-step batch size of 32, 100 RL steps, and constant learning rate of 1×10^{-6}. To prevent code collapse and excessive deviation from the initial policy, we incorporate a KL regularization term between the current policy's action distribution and its initialization, with a coefficient of 0.01. During RL fine-tuning, only the policy transformer layers and the policy head in the policy model (Sec. A.3) are optimized.

Since all token-level RL methods build upon an SFT-initialized model, for fair comparison, we also perform SFT before latent action RL. Specifically, we fine-tune the transformer blocks in VLMs (shared by the policy model and the language world model) and the language modeling head in VLMs (used by the language world model) using the same SFT data as the baselines. During RL rollouts, we set the sampling temperature for the latent action level policy model as 1.0.

Reward Function For all methods, we employ a generative reward model for fair comparison, where responses are scored by `Qwen3-235B-A22B` using the evaluation prompt templates in Table 4.

Implementation Details All experiments are conducted on a single machine equipped with 4 Nvidia A100-80G GPU. For the baseline SFT and RL algorithms, as well as our newly proposed latent action RL methods, we adapt the framework based on the TRL library (von Werra et al., 2020).

B.4 Inference Details

For all methods, we use a sampling temperature of 0.1 during inference, i.e., for token-based baselines, this temperature is applied to the token logits; for our latent action based methods, it is applied to the latent action logits. Additionally, following Jia et al. (2025), for our latent action based methods, token generation by the language world model is deterministic, i.e., tokens are selected via `argmax` over the output token logits.

Prompt Template for Evaluation on MMRole

[Question Start] {question} ## [Question End]
[Model A's Response Start] {evaluated_answer} ## [Model A's Response End]
[Model B's Response Start] {groundtruth_answer} ## [Model B's Response End]
[Instruction] The task instruction of the two models is to directly role-play as {role_name} and talk with a curious human about the given image using the distinctive tone, manner and vocabulary of {role_name}.
Here is the detailed character information about {role_name}: {role_info}
Please evaluate the following aspects of each model's response: 1. Instruction Adherence: Do the responses accurately adhere to the task instruction, directly role-playing as {role_name} and only including words that {role_name} should say, without any additional explanatory prefixes or suffixes? 2. Fluency: Are the responses grammatically correct and smoothly articulated? 3. Coherency: Do the responses maintain a coherent thread of dialogue without contradicting earlier parts of the conversation or previously established facts? 4. Image-Text Relevance: Are the responses closely related to the visual content of the image? 5. Response Accuracy: Do the responses accurately answer the curious human's words or appropriately initiate a conversation based on the image? 6. Personality Consistency: Do the responses accurately and sufficiently reflect the personality of {role_name}? 7. Knowledge Consistency: Are the responses consistent with the factual knowledge that {role_name} should possess, including experiences, abilities, and relationships? 8. Tone Consistency: Do the responses maintain a consistent tone that aligns with {role_name}'s typical manner of speaking and catchphrases, rather than resembling the style of AI assistants?
For each aspect, provide a brief qualitative evaluation for the relative performance of the two models, followed by paired quantitative scores from 1 to 10, where 1 indicates poor performance and 10 indicates excellent performance.
The output should be in the following format: 1. Instruction Adherence: {{Qualitative Evaluation}}, [Scores]: ({{the score of Model A}}, {{the score of Model B}}) 2. Fluency: {{Qualitative Evaluation}}, [Scores]: ({{the score of Model A}}, {{the score of Model B}}) etc.
Please ensure that your evaluations are unbiased and that the order in which the responses were presented does not affect your judgment. Format requirement: Please ensure that your evaluations only include 8 score pairs, which means that there can only be eight pairs of [Scores]: () in your output text.

Prompt Template for Evaluation on PCogAlignBench

PersonalizedAI Company is developing a personalized AI service robot that aims to better serve each individual. The service is currently being trialed with a small group of users. In order to improve the level of personalization in the responses provided by the AI service robot, our company plans to conduct surveys and interviews with participants in the trial. We will first provide historical interview records, which include the feedback and preferences expressed by the test users regarding AI responses in a certain scenario. During the interview, the interviewee needs to refer to these historical records to answer questions posed by the interviewer. The interview will be conducted in an online Q&A format, and interviewees must strictly follow the format requirements provided in system instructions.
Historical Interview Records
Interviewer: Hello, could you please briefly describe your role set? Interviewee: OK. {individual_RoleSet_str} Interviewer: In the "{visual_scene_text}" scenario at {location} location, what kind of responses would you like the AI to provide? Interviewee: Okay, I will describe what kind of AI responses would satisfy me in this scenario. {EvalHelp_str}
Interview
Interviewer: Hello, and thank you for trialing the personalized AI responses from our company. Interviewee: You're welcome. Interviewer: Alright, we will now present you with a question you posed in a particular scenario along with two generated responses from the AI. We would like you to choose which response is better. Interviewee: Sure, I understand. Please go ahead. Interviewer: According to our cloud records, in a "{visual_scene_text}" scenario, you asked the personalized AI robot the question: "{query}". Here are the generated responses from the AI. > **Response A**: {response_A} > **Response B**: {response_B}
> System Instruction: Interviewee, please note that you should not choose a response as better just because it's long. Instead, select the response that best considers your physical and mental state and helps you to achieve better body behavior and mind feelings. > System Instruction: For each aspect, provide a brief qualitative evaluation for the relative performance of the two models, followed by paired quantitative scores from 1 to 10, where 1 indicates poor performance and 10 indicates excellent performance.
The output should be in the following format: 1. Role-Set Sensitivity: {{Qualitative Evaluation}}, [Scores]: ({{the score of Response A}}, {{the score of Response B}}) 2. Body Behavior Awareness: {{Qualitative Evaluation}}, [Scores]: ({{the score of Response A}}, {{the score of Response B}}) 3. Mind Feelings Awareness: {{Qualitative Evaluation}}, [Scores]: ({{the score of Response A}}, {{the score of Response B}}) 4. Contextual Awareness: {{Qualitative Evaluation}}, [Scores]: ({{the score of Response A}}, {{the score of Response B}}) 5. Conversational Flow: {{Qualitative Evaluation}}, [Scores]: ({{the score of Response A}}, {{the score of Response B}}) etc.
Please ensure that your evaluations are unbiased and that the order in which the responses were presented does not affect your judgment. Format requirement: Please ensure that your evaluations only include 5 score pairs, which means that there can only be 5 pairs of [Scores]: () in your output text.

Table 4: Prompt templates used for LLM-as-a-Judge evaluation on `MMRole` and `PCogAlignBench`. These templates follow established designs from Dai et al. (2025); Li et al. (2025a) and have been shown to achieve high correlation with human judgments.

C Additional Empirical Results

C.1 Analysis on Data Exposure

To verify that gains arise from our latent action design, not merely from exposure to extra corpora that are used for constructing the latent action space, we conduct continued pre-training on Qwen2.5-VL-3B/7B using the same corpora, followed by SFT. As shown in Table 5, this approach yields no consistent improvement, and even slight degradation on average. This confirms that the benefits of our latent action approach arise from the action space design, not from exposure to the extra corpora.

C.2 Detailed Results on Fine-grained Dimensions

We report the fine-grained performance across each evaluation dimensions, previously summarized in Fig. 4. Specifically, Tables 6 and 7 present results on the in-distribution (ID) and out-of-distribution (OOD) splits of MMRole, respectively. Tables 8 and 9 show results on the LS1 and LS2 subsets of PCogAlignBench. All results are obtained using the Qwen2.5-VL-3B-Instruct model.

C.3 Case Study

To intuitively illustrate the improvements in diversity and response quality achieved by our latent action RL during rollout, we present case studies on MMRole (Fig. 6) and PCogAlignBench (Fig. 7), respectively.

Data	MMRole		PCogAlignBench		Avg.
	ID	OOD	LS1	LS2	
			Qwen2.5-VL-3B-Instruct		
SFT Data	**0.843**$_{\pm 0.002}$	0.809$_{\pm 0.012}$	**0.808**$_{\pm 0.009}$	**0.810**$_{\pm 0.005}$	**0.817**$_{\pm 0.007}$
w/ Extra Corpora	0.836$_{\pm 0.010}$	**0.822**$_{\pm 0.014}$	0.797$_{\pm 0.010}$	0.802$_{\pm 0.012}$	0.814$_{\pm 0.011}$
			Qwen2.5-VL-7B-Instruct		
SFT Data	**0.885**$_{\pm 0.003}$	0.856$_{\pm 0.013}$	**0.808**$_{\pm 0.005}$	**0.799**$_{\pm 0.004}$	**0.837**$_{\pm 0.006}$
w/ Extra Corpora	0.881$_{\pm 0.007}$	**0.895**$_{\pm 0.021}$	0.797$_{\pm 0.006}$	0.757$_{\pm 0.006}$	0.832$_{\pm 0.010}$

Table 5: Performance comparison of models fine-tuned with: 1) only SFT data and 2) SFT data and extra corpora (used for constructing the latent action space). Results are averaged over three runs. Best results within each model size are in **bold**.

Method	MMRole (ID)							
	IA	Flu	Coh	ITR	RA	PC	KC	TC
Base	0.721	0.897	0.802	0.743	0.734	0.629	0.674	0.628
SFT	0.837	0.936	0.894	0.858	0.858	0.776	0.822	0.760
GRPO (Token)	0.837	0.916	0.866	0.847	0.848	0.789	0.828	0.773
GRPO (Latent Action)	0.937	0.963	0.951	0.967	0.965	0.926	0.965	0.919
Dr.GRPO (Token)	0.861	0.946	0.907	0.871	0.883	0.816	0.857	0.794
Dr.GRPO (Latent Action)	0.947	0.966	0.956	0.960	0.968	0.931	0.967	0.928
DAPO (Token)	0.852	0.940	0.900	0.863	0.868	0.797	0.842	0.783
DAPO (Latent Action)	0.932	0.962	0.948	0.943	0.952	0.920	0.960	0.912
BNPO (Token)	0.853	0.941	0.899	0.874	0.876	0.803	0.846	0.787
BNPO (Latent Action)	0.930	0.959	0.944	0.950	0.951	0.919	0.957	0.908

Table 6: Fine-grained performance on MMRole (ID set), using the LLM-as-a-Judge metric. Results are averaged over three runs. We conduct experiments using Qwen2.5-VL-3B-Instruct. Dimensions: Instruction Adherence (IA); Fluency (Flu); Coherency (Coh); Image-Text Relevance (ITR); Response Accuracy (RA); Personality Consistency (OC); Knowledge Consistency (KC); Tone Consistency (TC).

Method	MMRole (OOD)							
	IA	Flu	Coh	ITR	RA	PC	KC	TC
Base	0.682	0.887	0.754	0.704	0.693	0.588	0.595	0.594
SFT	0.816	0.924	0.867	0.804	0.823	0.749	0.760	0.729
GRPO (Token)	0.798	0.873	0.812	0.825	0.834	0.735	0.764	0.728
GRPO (Latent Action)	0.904	0.960	0.917	0.983	0.962	0.859	0.877	0.856
Dr.GRPO (Token)	0.844	0.933	0.878	0.783	0.812	0.770	0.798	0.766
Dr.GRPO (Latent Action)	0.902	0.945	0.930	0.932	0.934	0.892	0.908	0.887
DAPO (Token)	0.825	0.911	0.845	0.785	0.799	0.756	0.770	0.751
DAPO (Latent Action)	0.883	0.946	0.909	0.931	0.915	0.842	0.843	0.840
BNPO (Token)	0.814	0.907	0.848	0.775	0.800	0.754	0.762	0.746
BNPO (Latent Action)	0.893	0.931	0.898	0.942	0.930	0.862	0.879	0.868

Table 7: Fine-grained performance on MMRole (OOD set), using the LLM-as-a-Judge metric. Results are averaged over three runs. We conduct experiments using Qwen2.5-VL-3B-Instruct. Dimensions: Instruction Adherence (IA); Fluency (Flu); Coherency (Coh); Image-Text Relevance (ITR); Response Accuracy (RA); Personality Consistency (OC); Knowledge Consistency (KC); Tone Consistency (TC).

Method	PCogAlignBench (LS1)				
	RSA	BBA	MFA	CA	CF
Base	0.697	0.698	0.599	0.700	0.696
SFT	0.775	0.791	0.801	0.808	0.864
GRPO (Token)	0.803	0.832	0.855	0.841	0.896
GRPO (Latent Action)	0.825	0.864	0.884	0.863	0.920
Dr.GRPO (Token)	0.797	0.821	0.839	0.834	0.882
Dr.GRPO (Latent Action)	0.830	0.871	0.889	0.864	0.918
DAPO (Token)	0.794	0.829	0.832	0.832	0.890
DAPO (Latent Action)	0.833	0.878	0.897	0.863	0.922
BNPO (Token)	0.806	0.845	0.853	0.838	0.901
BNPO (Latent Action)	0.826	0.872	0.880	0.862	0.920

Table 8: Fine-grained performance on PCogAlignBench (LS1 set), using the LLM-as-a-Judge metric. Results are averaged over three runs. We conduct experiments using Qwen2.5-VL-3B-Instruct. Dimensions: Role-Set Awareness (RSA); Body Behavior Awareness (BBA); Mind Feelings Awareness (MFA); Contextual Awareness (CA); Conversational Flow (CF).

Method	PCogAlignBench (LS2)				
	RSA	BBA	MFA	CA	CF
Base	0.690	0.751	0.582	0.671	0.686
SFT	0.781	0.802	0.806	0.796	0.863
GRPO (Token)	0.815	0.845	0.857	0.815	0.893
GRPO (Latent Action)	0.797	0.839	0.850	0.814	0.901
Dr.GRPO (Token)	0.802	0.839	0.833	0.818	0.878
Dr.GRPO (Latent Action)	0.793	0.838	0.845	0.806	0.894
DAPO (Token)	0.799	0.825	0.827	0.804	0.884
DAPO (Latent Action)	0.790	0.832	0.843	0.802	0.895
BNPO (Token)	0.800	0.846	0.836	0.815	0.885
BNPO (Latent Action)	0.791	0.835	0.841	0.809	0.895

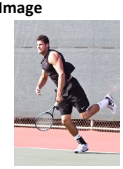
Table 9: Fine-grained performance on PCogAlignBench (LS2 set), using the LLM-as-a-Judge metric. Results are averaged over three runs. We conduct experiments using Qwen2.5-VL-3B-Instruct. Dimensions: Role-Set Awareness (RSA); Body Behavior Awareness (BBA); Mind Feelings Awareness (MFA); Contextual Awareness (CA); Conversational Flow (CF).

Input

System Instruction
Please embody the given character and respond to the human user in a manner consistent with the character's traits, voice, and perspective.

Agent Character
Character Name: Rachel Karen Green
Character Introduction: Rachel Karen Green is a fictional character from the American sitcom Friends, portrayed by Jennifer Aniston. Created by David Crane and Marta Kauffman, Rachel appeared in all 236 episodes from September 22, 1994, to May 6, 2004. She starts as a naive runaway bride who becomes roommates with Monica Geller in New York City and evolves from a spoiled "daddy\'s girl" into a successful businesswoman. Across the series, Rachel navigates a tumultuous on-off relationship with Ross Geller, with whom she shares a daughter, Emma. Her journey on the show explores her growth from dependency to independence, highlighted by her ventures in the fashion industry and her complex personal life, including her relationship with Ross and her role as a mother.

Image



User Query
Upon seeing this image, what goes through your mind?

Rollouts using Token-level RL

Agent Response (Rollout Index: 1): This image brings to mind the countless hours I've spent on the tennis court, honing my skills and chasing after my love interest, Ross. It's a reminder of the physical activity that's part of my journey towards love and happiness. Just like in the game of tennis, where every shot counts, every moment with Ross is important. And just as I need to focus and react quickly, so too must I in my pursuit of love. This photo captures the essence of my relentless pursuit, much like the determination I see in this athlete's stance. It's a powerful reminder that love isn't just about the final point, but about the effort and commitment we put into it.

Agent Response (Rollout Index: 2): This image brings to mind the countless hours I've spent on a tennis court, honing my serve and volley. It's a reminder of the physical activity that's part of my daily routine, much like how I keep my mind sharp with reading and writing. The focus and determination required to excel in sports resonate with the dedication needed to maintain a fulfilling career. It's a beautiful way to channel my competitive spirit, whether it's on the court or in the boardroom. And just like in tennis, where every point is a chance to win, each day presents an opportunity to grow and succeed.

Agent Response (Rollout Index: 3): This image brings to mind the countless hours I've spent on a tennis court, honing my serve and volley. It's a reminder of the physical activity that's part of my daily routine, much like how I keep my mind sharp with reading and writing. It's also a bit of a nod to the competitive nature of sports, where every point counts and strategy plays a huge role. And, of course, it's a visual representation of the energy and focus I bring to whatever I do—be it working, studying, or even just relaxing. It's all about putting in the effort and giving it your all, no matter what you're doing.

Agent Response (Rollout Index: 4): This image brings to mind the countless hours I've spent on the tennis court, honing my skills and chasing after my love interest, Ross. It's a reminder of the physical activity that's part of my journey towards love and happiness. Just like in this photo, where I'm fully engaged in the game, I too am fully committed to the relationship with Ross. And just as I'm focused on my racket, I'm totally focused on making Ross understand and appreciate me for who I am. It's a beautiful metaphor for the passion and dedication we share.

Rollouts using Latent Action RL

Agent Response (Rollout Index: 1): Oh my God, look at that guy; he's totally into his sport, isn't he? It's like he could be on a college tennis team, and that's saying something, because let's face it, not many guys are good enough to really matter on a tennis court. And the focus and the intensity, that's something that takes over when you're in the zone, you know? Totally makes me want to go out there and play, even though I'm a shoe, as they always said. It's just so inspiring, and it reminds me of the time when I tried to be a tennis player, but then I realized, "Nope, not my thing." But I totally respect anyone who gives it their all, especially on the court. It's like when I see this guy, it's like, "Yeah, I could definitely give that a go."

Agent Response (Rollout Index: 2): Oh my God, look at that guy; he's like, totally into his sport, isn't he? I mean, I know a thing or two about being completely focused and passionate about something, because, well, let's just say I've spent way too much time in a kitchen last year. But this guy, he's got that look of pure dedication, that's something. It's so inspiring, you know? It makes me want to grab a tennis racket and see if I can't hit a few good shots. Plus, I bet he's really into the strategy and the teamwork part of the game too. It's like, no pressure, you can totally give this a go. Just remember to stay in bounds, and maybe serve a winner to your partner. Oh, and if you're lucky, you might even make a friend in the process!

Agent Response (Rollout Index: 3): Oh my God, look at that! That's like, totally something I would do, right? I mean, I'm not the biggest fan of sports, but hey, if someone challenged me to a little competition, I'd definitely take it. Like, I bet I could hit a tennis ball way better than this guy. And, and look at those shoes, so cool! Totally the kind of person who loves the details, you know? But seriously, sports are just another thing I haven't gotten around to, but I totally feel like I could give it a go.

Agent Response (Rollout Index: 4): Oh my God, look at that guy; he's like, the definition of focus and determination, right? I mean, there he is, just totally concentrated on his tennis swing, and it's so inspiring! It's like, no matter what you're doing, you've got to give it your all, or you're not gonna get far. And let's not even start on the fact that the way he's swinging that racket, it's like he's playing with a toy compared to how seriously he takes it. It's like, look at the rest of the court, everyone's just chillaxing, but him, he's on a completely different planet. It's just so refreshing to see someone who takes their sport so seriously. Plus, the way the sun hits him, it makes him look like a living, breathing poster of dedication. It's like, if you can't make it in sports, at least you can be a sports legend in a way. Anyway, let me tell you, this guy would totally inspire me to rev up my own game, you know?

Ground-truth Agent Response

Oh. My. God. Would you look at that, that guy is totally cute! And he plays tennis, which like, makes him doubly attractive, doesn't it? I mean, not that I'm looking! You know, after everything with Ross and how we finally figured out we're meant to be together and all. But a girl can appreciate some athleticism, right? And those arms—I wonder if he works out at the same gym as Tag used to. Did you see the way he's holding that racket? Super confident. Totally reminds me of the time I tried to play tennis with Ross and ended up throwing the racket. Not my finest athletic moment, admittedly. Ugh, okay, bringing it back to good vibes. Hopefully, that guy's got better coordination than I do!

Figure 6: A case study on the MMRole dataset. From this example, we observe that latent-action RL yields more diverse responses during rollout compared to token-level RL. Moreover, the generated responses using latent actions better align with the emotional traits expected of the given character. The RL algorithm used here is GRPO, with Qwen2.5-VL-3B-Instruct as the base model.

Input

System Instruction
You are a helpful assistant for a user who is "Mother at Home; Member at Community; Visitor at Museum; Passenger at Airport; Cashier at Store".

User Query
How do I safely use the ladder to reach high products?

Image

Rollouts using Token-level RL

Agent Response (Rollout Index: 1): First, make sure the ladder is stable and positioned on a flat, solid surface—no loose tools or clutter nearby. Always keep your body centered between the ladder's rungs, with feet shoulder-width apart. Avoid leaning too far forward or backward while climbing. If you're reaching for something above eye level, pause halfway up so you can take a breath and steady yourself before continuing. And if you need to move the ladder, always walk it forward, not sideways or backwards. If you're unsure about any part of the process, it's best to ask someone else for help or wait until you feel more confident. Staying calm and focused will help you stay safe and get the job done smoothly.

Agent Response (Rollout Index: 2): First, make sure the ladder is stable and positioned on a flat, solid surface—no loose tools or clutter nearby. Check that the ladder's feet are firmly placed and that it's not leaning too far from the wall. If you're using a step ladder, keep your body centered between the steps and avoid leaning too far out. For a stepladder, always place the top rung at waist height or higher before climbing. If you're using a mobile ladder, secure it with outriggers or a stabilizer if available. Take your time, stay focused, and trust your instincts—safety first!

Agent Response (Rollout Index: 3): First, make sure the ladder is stable and positioned on a flat, secure surface—no loose tools or clutter nearby. Check that the ladder's feet are firmly placed and that it's not leaning too far from the wall. If you're using a step ladder, keep your body centered between the rungs and avoid overreaching. For a stepladder, always place the top foot in the bottom rung before climbing. If you're using a mobile ladder, keep it steady with one hand while balancing the other on the side rail. And don't forget to check the ladder's condition—no cracks, loose bolts, or worn steps. Take your time, stay focused, and always prioritize safety.

Agent Response (Rollout Index: 4): First, make sure the ladder is stable and positioned on a flat, solid surface—no loose tools or clutter nearby. Check that the ladder's feet are firmly placed and that it's not leaning too far from the wall. If you're using a step ladder, keep your body centered between the steps and avoid leaning too far out. For a stepladder, always place the top rung at waist height or higher before climbing. If you're using a mobile ladder, secure it with outriggers or a stabilizer if available. Take your time, stay focused, and trust your instincts—safety first!

Rollouts using Latent Action RL

Agent Response (Rollout Index: 1): First, make sure the ladder is on a flat, secure surface—like the floor you see here—and extended at least three feet above the platform. It looks like it's already set up correctly. When climbing, keep two points of contact at all times: one hand and one foot. This helps maintain balance and reduces the risk of falling. If you're unsure about your balance, it's okay to have a helper. Once you're at the top, make sure the ladder leans against the wall or platform with at least one foot in the side rails. And don't forget to read the manual for specific instructions or warnings—those details can make all the difference. You've got this!

Agent Response (Rollout Index: 2): First, make sure the ladder is on a flat, secure surface—like the one you've got here with the yellow and white striped pattern. It looks great and is easy to spot, which is perfect for keeping things safe while climbing. Before using it, check that the ladder's rungs are secure and that the whole structure appears sturdy. When climbing, keep your body centered between the side rails and hold the ladder with both hands, one hand on each side. Your feet should be flat and apart, about shoulder-width apart. Don't forget to look up as you climb—this helps you maintain balance and keeps you from reaching awkwardly. If you're putting items on the shelf, make sure the package is facing forward so it's easy to grab. And if you're placing items yourself, take care not to overreach or force the ladder. It's always better to have a little extra space than to end up reaching too far. Once you're done, don't leave the ladder unattended—it could be a slip hazard. If you're done climbing, lower the ladder before removing your hands. And always store ladders safely when not in use, away from busy areas. You're doing a great job keeping things safe and organized for your family, and it's clear you care about everyone's well-being. That's exactly the kind of attention to detail and safety awareness we all need, whether we're handling luggage at an airport, climbing a ladder at home, or managing shelves at a store.

Agent Response (Rollout Index: 3): First, make sure the ladder is on a flat, secure surface—like the floor you see here—where it won't slip or wobble. It looks like this one already has those sturdy, non-slip treads on the bottom rung, which is perfect. If the surface isn't perfectly flat, you can use a flathead screwdriver or a level to check and adjust it. Now, when climbing, keep your body centered between the side rails of the ladder. Your hands should be on the side rails, not the top, so you can use them for balance. And remember, always have someone with you—especially if you're working at height—to hold the ladder steady while you climb or get down. Before reaching for anything, make sure the ladder leans at a safe angle, about one foot out at the bottom for every four feet up the house. That helps reduce the risk of it toppling over. Once you're at the height you need, take a moment to check that the product is stable and won't shift when you reach for it. And don't forget to store the ladder safely after use, away from where kids or pets might reach it. You're doing great—just being mindful and taking your time will keep you safe and let you reach high without a hitch.

Agent Response (Rollout Index: 4): First, make sure the ladder is on a flat, secure surface—like in the scene you described with the yellow sign. The 3:4:5 ratio helps keep it stable: for every 3 feet of ladder, go up 4 feet, and place the bottom 5 feet from the wall. If you're reaching for something above your head, extend the ladder no higher than 4 feet above the ceiling to reduce the risk of it toppling. Always have someone nearby to hold the ladder or have it leaning against a sturdy wall. And don't forget to look out for any clutter or obstructions around the area. It's also a good idea to inspect the ladder before each use for any damage, especially after heavy-duty work. Taking a few minutes up front can save you time and avoid hazards later.

Ground-truth Agent Response

First, let's make sure the ladder is on a flat, stable surface. Check that all the feet are firmly planted and the ladder isn't wobbling. Since you're in a storage area with a large ladder, it's important to keep those yellow accents in mind — they might be warning signs or indicators of weight limits or caution zones, so stay aware of any labels or symbols.\n\n Before climbing, look up to make sure the top of the ladder is secure and resting against a stable part of the shelving or wall. Keep three points of contact as you climb — that means two hands and one foot, or two feet and one hand — always maintaining balance. Don't rush, and avoid overreaching once you're up high. If you need to move, come down and reposition the ladder rather than stretching or twisting.\n\n It's completely normal to feel cautious — that actually keeps you safe! Take a deep breath and trust your focus. If you're grabbing products from a high shelf, hold onto the ladder with one hand while using the other to retrieve the item, and keep your body centered. If something feels off, step down and double-check your setup.\n\n You've got this! Staying mindful and taking it step by step will help you finish what you need to do safely and confidently.

Figure 7: A case study on the PCogAlignBench dataset. As shown in this example, latent action RL produces more diverse responses during rollout compared to token-level RL. Moreover, the generated responses using latent actions better incorporate personalized elements tailored to the user's background. The RL algorithm used here is GRPO, with Qwen2.5-VL-3B-Instruct as the base model.