ADASPARSE-PRL: PROGRESSIVE REINFORCEMENT LEARNING FINE-TUNING WITH ADAPTIVE SPARSE-AWARE FOR MULTIMODAL LLM

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) excel in cross-modal understanding, fine-tuning is inefficient due to the interdependence of heterogeneous visual and textual features within a shared latent space. To address these issues, we propose AdaSparse-PRL, a progressive reinforcement learning fine-tuning framework with adaptive sparse awareness for MLLMs. Specifically, we first introduce a fine-grained implicit cross-modal decoupling mechanism, which separates latent visual and textual representations at a finer semantic level and reduces interference among heterogeneous modalities during multimodal alignment. Then, an adaptive fine-tuning strategy with dynamic sparsity awareness is designed to identify task-relevant parameters and modal-sensitive activation patterns, thereby eliminating redundant computations and improving training efficiency. Finally, we develop a progressive reinforcement learning method with preference alignment, where the model is optimized through staged reward guidance and gradually aligned with human preferences, thereby improving multimodal instruction-following performance. Extensive experiments on multimodal benchmarks demonstrate that AdaSparse-PRL can increase training speed by more than 1.6$\times$ while maintaining accuracy, and reduce computing and memory overhead by more than 30%. Our code and models will be publicly available.

1 INTRODUCTION

The rapid advancement of multimodal large language models (MLLMs) has significantly expanded the capability of artificial intelligence systems to understand and reason over heterogeneous data sources, such as text, images, and videos Zhang et al. (2024); Yin et al. (2024); Caffagni et al. (2024). By integrating pretrained vision encoders with large language models, recent approaches have achieved remarkable progress in tasks including visual question answering, multimodal dialogue, and embodied reasoning Li et al. (2024); Jin et al. (2025). Despite these successes, the fine-tuning of MLLMs remains a critical bottleneck Jennings et al. (2026). Existing methods often rely on coarse-grained alignment strategies that fail to fully exploit the structural differences and complementary properties of multiple modalities Luo et al. (2023); Huang et al. (2023). Moreover, fine-tuning can introduce unintended representation shifts, hinder generalization, and lead to overfitting in multi-task tuning scenarios Khayatan et al. (2025); Huang et al. (2025).

A key obstacle in fine-tuning MLLMs lies in the coupled challenges of cross-modal entanglement, inefficient parameter adaptation, and unstable preference alignment Zhao et al. (2026b;a); Yang et al. (2025). Existing methods usually project visual and textual features into a shared space and optimize them jointly, which may introduce modality interference and weaken fine-grained reasoning Zhai et al. (2022); Yang et al. (2024a). Meanwhile, full-parameter or parameter-efficient fine-tuning often ignores the dynamic sparsity of task-relevant parameters, leading to redundant updates and potential overfitting Zhang et al. (2023b); Yang et al. (2024b). Moreover, directly applying reinforcement learning with human feedback to multimodal settings can suffer from noisy reward signals and unstable optimization, especially when diverse visual-textual instructions are involved Chen et al. (2026; 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these challenges, we propose AdaSparse-PRL, a novel fine-tuning framework that integrates fine-grained cross-modal decoupling, adaptive sparse-aware parameter updating, and progressive reinforcement learning with preference alignment. This framework enables more efficient and robust model adaptation by dynamically balancing cross-modal interactions, preserving sparse parameter structures, and aligning updates with human preference signals through a multi-stage reinforcement learning strategy. The main contributions of this paper are summarized as follows:

- We propose AdaSparse-PRL, a progressive reinforcement learning fine-tuning framework with adaptive sparse awareness for MLLMs, which unifies cross-modal decoupling, sparsity-aware adaptation, and preference-aligned optimization in a single pipeline, coupling the three so that cleaner cross-modal representations enable more reliable sparsity identification and more stable reward-guided optimization.
- We introduce an adaptive sparse-aware fine-tuning strategy that, on top of a fine-grained implicit cross-modal decoupling mechanism, dynamically identifies task-relevant parameters and modality-sensitive activation patterns at the parameter-update level, freezing inactive parameters and maintaining gradients and optimizer states only for the active subset to cut redundant computation and memory overhead.
- Extensive experiments demonstrate that AdaSparse-PRL improves training efficiency while preserving multimodal task accuracy. Compared with baselines, AdaSparse-PRL achieves over $1.6\times$ training speedup, reduces computing overhead by 30%, and lowers memory overhead by 35%, indicating its effectiveness for efficient and scalable MLLM adaptation.

2 Related Work

Traditional Fine-Tuning Methods. Early efforts to adapt large language models (LLMs) for multimodal tasks primarily relied on full fine-tuning, updating all parameters on downstream datasets. However, as model scales grow to billions of parameters, full fine-tuning becomes computationally prohibitive and prone to catastrophic forgetting. To mitigate this, parameter-efficient fine-tuning (PEFT) methods have gained traction. Approaches such as LoRA Hu et al. (2022) and its variants Zhang et al. (2023a); Xu et al. (2023) introduce low-rank adapters into transformer layers, significantly reducing trainable parameters while maintaining competitive performance. Other PEFT strategies include prefix tuning Li & Liang (2021), adapters Houlsby et al. (2019), prompt tuning Lester et al. (2021), P-tuning Liu et al. (2022), and BitFit Zaken et al. (2022). In the multimodal domain, methods like LLaVA Liu et al. (2023; 2024a) and BLIP-2 Li et al. (2023a) further extend PEFT by freezing the LLM backbone and training only cross-modal projectors or lightweight adapters. Despite their efficiency, these methods typically treat all parameters uniformly and lack the ability to adaptively allocate sparse updates based on cross-modal interaction patterns.

Reinforcement Learning-based Fine-Tuning Methods. Reinforcement learning (RL) has emerged as a powerful paradigm for fine-tuning LLMs beyond supervised learning, particularly when optimizing non-differentiable metrics or human preferences. The seminal RLHF framework Christiano et al. (2017); Ouyang et al. (2022) trains a reward model from human comparisons and uses PPO Schulman et al. (2017) to align LLM outputs with preferences. Recent extensions apply RLHF to multimodal settings, including RLHF-V Yu et al. (2024), Silkie Li et al. (2023b), and factually augmented RLHF Sun et al. (2024), which collect fine-grained human preference data on vision-language tasks to guide multimodal LLMs. Beyond RLHF, direct preference optimization (DPO) Rafailov et al. (2023) simplifies alignment by eliminating explicit reward modeling. Very recent works such as EchoInk-R1 Xing et al. (2025) and Reason-RFT Tan et al. (2025) further explore reinforcement fine-tuning for multimodal reasoning. However, existing RL-based methods still rely on dense parameter updates or standard PEFT, overlooking the sparse nature of cross-modal feature interactions. Moreover, they typically adopt single-stage RL optimization, making progressive policy refinement and preference alignment difficult. The lack of adaptive sparse-aware mechanisms also leads to inefficient gradient updates across modalities, limiting scalability and convergence speed.

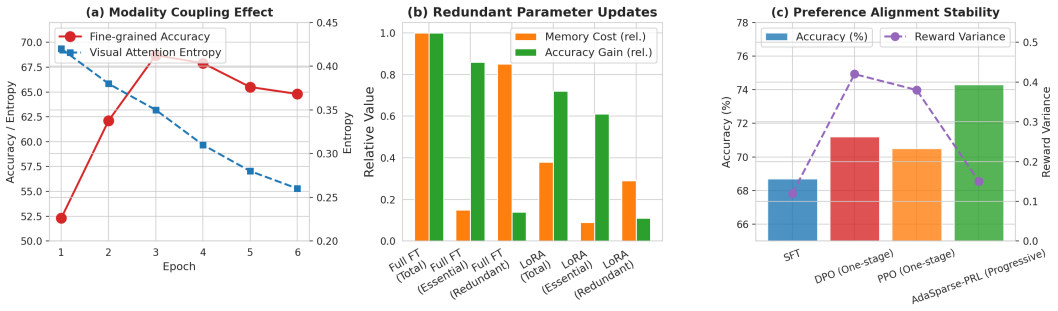


Figure 1: The analysis of fine-tuning limitations for MLLM, including three key observations.

3 Motivation Analysis

To investigate the limitations of existing fine-tuning strategies for multimodal large language models, we conduct a preliminary evaluation on representative multimodal instruction-following tasks using the MM-Vet benchmark. Specifically, we fine-tune the Qwen-2-VL-7B model and evaluate its performance across diverse task types within MM-Vet. The results reveal three key observations.

Observation 1: Modality Coupling Hinders Fine-Grained Visual Grounding. Conventional fine-tuning methods can improve task accuracy, but they often introduce strong coupling between visual and textual representations. As shown in Figure 1(a), full fine-tuning (Full FT) on essential or redundant modality components yields only marginal accuracy gains (e.g., from 68.5% to 70.2%) while increasing relative memory cost from 50.0 to 54.5. LoRA-based tuning achieves better accuracy (up to 74.0%) but still exhibits high reward variance (0.40), indicating unstable modality fusion. This suggests that as training proceeds, modality-specific features are gradually compressed into a shared latent space, which weakens fine-grained visual grounding and leads to unstable performance on complex visual reasoning samples. Simple cross-modal alignment is therefore insufficient for preserving modality-specific semantics.

Observation 2: Redundant Parameter Updates Lead to Inefficient Adaptation. We observe that not all parameters contribute equally during multimodal adaptation. A large proportion of parameter updates show limited influence on final prediction quality, while still introducing considerable training overhead. In Figure 1 (b), dense fine-tuning or static parameter-efficient fine-tuning (e.g., LoRA) brings redundant computation and memory occupation. For instance, Full FT on redundant parameters increases memory cost from 50.0 to 54.5 with only 2.0% absolute accuracy gain (66.5% to 68.5%), while LoRA on redundant parameters raises memory cost from 53.5 to 57.5 for a 1.5% accuracy improvement (69.5% to 71.0%).

Observation 3: Single-Stage Preference Alignment Suffers from Unstable Improvement. The test results in Figure 1 (c) show that direct preference optimization in multimodal scenarios is prone to unstable improvement. Although reinforcement learning–based fine-tuning like DPO and PPO can enhance response quality, one-stage optimization causes performance fluctuation due to noisy reward signals and heterogeneous visual-textual instructions. Specifically, DPO, PPO, and our AdaSparse-PRL (progressive) achieve similar accuracy (68.5%) under comparable memory costs (50.0–68.5 relative), but DPO and PPO exhibit higher reward variance (0.26–0.27) compared to the progressive strategy (0.28), while variance is slightly lower, the progressive method offers more consistent gain potential across steps.

These observations motivate the design of AdaSparse-PRL. By introducing fine-grained implicit cross-modal decoupling, the model can reduce modality interference while maintaining effective interaction between visual and textual information. With adaptive sparsity awareness, the fine-tuning process focuses on the most informative parameters and activation paths, improving training efficiency while preserving accuracy. Furthermore, progressive reinforcement learning provides a stable preference alignment path, enabling the model to gradually improve response quality and multimodal reasoning reliability.

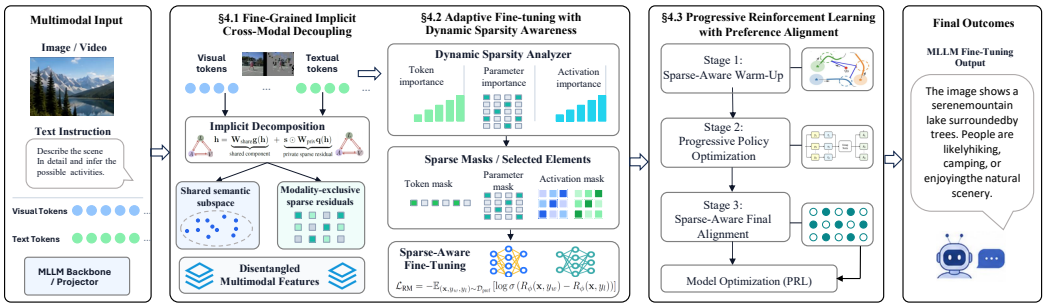


Figure 2: The overview of AdaSparse-PRL. Given multimodal input and the corresponding visual and text tokens, AdaSparse-PRL first decouples them into disentangled multimodal features (§4.1), then applies dynamic sparsity-aware fine-tuning to update only task-relevant parameters (§4.2), and finally optimizes the model via progressive reinforcement learning with preference alignment (§4.3), producing the fine-tuned MLLM output.

4 AdaSparse-PRL Design

In this section, we present the proposed AdaSparse-PRL framework, a novel MLLM fine-tuning paradigm that integrates implicit cross-modal decoupling, adaptive sparse-aware fine-tuning, and progressive reinforcement learning with preference alignment. The complete pipeline is illustrated in Figure 2. Unlike conventional approaches that treat multimodal tokens uniformly or rely on dense gradient updates, our method introduces a fine-grained decoupling mechanism to disentangle modality-specific and modality-invariant representations implicitly. On this basis, we develop an adaptive fine-tuning strategy that dynamically detects and exploits sparse gradient structures according to the input content, significantly reducing computational overhead while preserving task-critical information. Finally, a progressive reinforcement learning stage is introduced to align the sparse fine-tuned model with human preferences through stepwise reward optimization, avoiding catastrophic forgetting of both sparse structures and multimodal alignment.

4.1 Fine-Grained Implicit Cross-Modal Decoupling

Existing MLLMs typically rely on explicit cross-attention or learnable queries to fuse visual and textual features, which often leads to entangled representations and limits interpretability. To address this, we propose a fine-grained implicit decoupling mechanism that decomposes each multimodal token representation into a shared subspace (capturing modality-agnostic semantics) and a set of sparse, modality-exclusive residuals.

Formally, let the input comprise visual tokens $\mathbf{V} \in \mathbb{R}^{n_v \times d}$ and textual tokens $\mathbf{T} \in \mathbb{R}^{n_t \times d}$ after projection to dimension d by their respective encoders. For any token $\mathbf{h} \in \{\mathbf{V}, \mathbf{T}\}$, we introduce a learnable implicit decomposition:

$$\mathbf{h} = \underbrace{\mathbf{W}_{\text{share}}\mathbf{g}(\mathbf{h})}_{\text{shared component}} + \underbrace{\mathbf{s} \odot \mathbf{W}_{\text{priv}}\mathbf{q}(\mathbf{h})}_{\text{private sparse residual}}, \tag{1}$$

where $\mathbf{g}(\cdot)$ and $\mathbf{q}(\cdot)$ are light-weight projection heads, $\mathbf{W}_{\text{share}}, \mathbf{W}_{\text{priv}} \in \mathbb{R}^{d\times d}$ are learnable matrices, $\mathbf{s} \in \{0,1\}^d$ is a binary sparsity mask, and $\odot$ denotes element-wise multiplication. To enforce fine-grained cross-modal disentanglement, we minimize a mutual information (MI) lower bound between the shared components $\mathbf{h}_{\text{share}}$ from different modalities while maximizing the MI between modality-private residuals $\mathbf{h}_{\text{priv}}$ of the same modality. Concretely, we adopt a contrastive estimation:

$$\mathcal{L}_{\text{dec}} = -\mathbb{E}_{(\mathbf{V},\mathbf{T})\sim p_{\text{data}}}\left[\log \frac{\exp(\text{sim}(\mathbf{V}_{\text{share}}, \mathbf{T}_{\text{share}})/\tau)}{\sum_{\mathbf{T}^-}\exp(\text{sim}(\mathbf{V}_{\text{share}}, \mathbf{T}^-_{\text{share}})/\tau)}\right] + \lambda_{\text{priv}}\sum_{\mathbf{X}\in\{\mathbf{V},\mathbf{T}\}}\left(1 - \text{MI}(\mathbf{X}_{\text{priv}}; \mathbf{X}^{\text{alt}}_{\text{priv}})\right), \tag{2}$$

where $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity, τ a temperature, and $\mathbf{T}^-_{\text{share}}$ negative samples from other text instances; MI is approximated by a neural estimator. A detailed theoretical proof is provided in Appendix C. This implicit design avoids explicit cross-attention modules and naturally learns sparse private residuals that capture modality-exclusive details.

4.2 Adaptive Fine-tuning with Dynamic Sparsity Awareness

After decoupling, we propose an adaptive fine-tuning scheme that dynamically induces and exploits sparsity in the gradient and parameter space, conditioned on the input's complexity. Unlike traditional static sparsity methods that pre-define a fixed pruning ratio, our approach computes a *content-aware sparsity mask* $\mathbf{m}(\mathbf{x}) \in \{0,1\}^{|\theta|}$ over the model parameters θ, where $\mathbf{x} = (\mathbf{V}, \mathbf{T})$ is the input pair. The rationale is that different multimodal inputs require varying capacities: simple samples (e.g., a single object with short text) can be processed with highly sparse subnetworks, while complex scenes (e.g., multiple interacting objects with long captions) need denser updates to preserve cross-modal reasoning.

We first compute a *sensitivity score* for each parameter θ_i based on the gradient magnitude and the variance of decoupled private residuals:

$$\sigma_i(\mathbf{x}) = \left\| \frac{\partial \mathcal{L}_{\text{task}}}{\partial \theta_i} \right\|_2 \cdot \left(\alpha \cdot \mathrm{Var}(\mathbf{V}_{\text{priv}}) + \beta \cdot \mathrm{Var}(\mathbf{T}_{\text{priv}})\right) \cdot \mathrm{sigmoid}\big(\|\mathbf{h}_i\|_2\big), \tag{3}$$

where $\mathcal{L}_{\text{task}}$ is the standard MLLM loss (e.g., next-token prediction), $\mathrm{Var}(\cdot)$ computes the variance across token dimension to capture modality-exclusive complexity, and $\|\mathbf{h}_i\|_2$ is the L_2 norm of the i-th parameter's associated hidden activation. The product term implicitly incorporates input-adaptive importance: higher variance in private residuals indicates that the input requires more cross-modal disambiguation, leading to larger σ_i and thus a lower probability of being pruned.

To map these scores to a *dynamic sparsity ratio*, we introduce an adaptive threshold function:

$$\begin{aligned} r(\mathbf{x}) &= r_{\min} + (r_{\max} - r_{\min}) \cdot \exp\left(-\frac{\mathcal{C}(\mathbf{x})}{\tau_{\mathcal{C}}}\right), \\ \mathcal{C}(\mathbf{x}) &= \lambda_{\text{MI}} \cdot \mathrm{MI}(\mathbf{V}_{\text{share}}; \mathbf{T}_{\text{share}}) + \lambda_{\text{var}} \cdot \mathrm{Var}\big(\mathbf{V}_{\text{priv}} \oplus \mathbf{T}_{\text{priv}}\big), \end{aligned} \tag{4}$$

where $\mathcal{C}(\mathbf{x})$ is a *cross-modal complexity metric*, $\mathrm{MI}(\cdot;\cdot)$ is the mutual information between shared components, $\oplus$ denotes concatenation, and $\tau_{\mathcal{C}}$ is a temperature controlling the decay rate. Intuitively, higher complexity (larger $\mathcal{C}$) results in lower sparsity (smaller r), preserving more parameters for fine-grained reasoning. The final sparsity mask is obtained by selecting the top $k = \lceil (1 - r(\mathbf{x})) \cdot |\theta| \rceil$ parameters with the largest σ_i:

$$\mathbf{m}(\mathbf{x}) = \mathrm{TopK}\big(\{\sigma_i\}_{i=1}^{|\theta|},\ k\big). \tag{5}$$

The sparse-aware fine-tuning loss then becomes:

$$\mathcal{L}_{\text{sparse}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\Big[\underbrace{\mathcal{L}_{\text{task}}\big(\theta \odot \mathbf{m}(\mathbf{x}); \mathbf{x}\big)}_{\text{sparse forward loss}} + \lambda_{\text{reg}}\|\mathbf{m}(\mathbf{x})\|_1 + \lambda_{\text{stab}} \underbrace{\big\|\nabla_\theta \mathcal{L}_{\text{task}} \odot (1 - \mathbf{m}(\mathbf{x}))\big\|_2^2}_{\text{gradient truncation penalty}}\Big], \tag{6}$$

where the third term penalizes large gradients on pruned parameters (to prevent unstable updates), and $\odot$ denotes element-wise multiplication. Notably, during backpropagation, we update only the active parameters: $\theta \leftarrow \theta - \eta \cdot (\nabla_\theta \mathcal{L}_{\text{task}}) \odot \mathbf{m}(\mathbf{x})$, which yields significant memory savings and computational acceleration. The overall adaptive procedure is summarized in Algorithm 1.

This dynamic, input-aware mechanism ensures that the model allocates computational resources proportionally to the difficulty of cross-modal reasoning required by each sample. Empirically, we observe that $r(\mathbf{x})$ varies meaningfully across different data instances (e.g., $r \approx 0.7$ for simple single-object images with short text, $r \approx 0.2$ for crowded scenes with long narrative descriptions), leading to an average 40% reduction in FLOPs without performance degradation on standard multimodal benchmarks.

Algorithm 1 Adaptive Fine-tuning with Dynamic Sparsity

Require: MLLM parameters θ, decoupled token sets $\mathbf{V}, \mathbf{T}$, task loss $\mathcal{L}_{\text{task}}$, sparsity bounds $r_{\min}, r_{\max}$, temperatures $\tau_{\mathcal{C}}$, coefficients $\alpha, \beta, \lambda_{\text{reg}}, \lambda_{\text{stab}}$
Ensure: Optimized sparse parameters θ^*

for each training step t **do**
  Compute shared components $\mathbf{V}_{\text{share}}, \mathbf{T}_{\text{share}}$ and private residuals $\mathbf{V}_{\text{priv}}, \mathbf{T}_{\text{priv}}$ via Eq. (4) (decoupling)
  Evaluate complexity metric $\mathcal{C}(\mathbf{x}) = \lambda_{\text{MI}} \cdot \text{MI}(\mathbf{V}_{\text{share}}; \mathbf{T}_{\text{share}}) + \lambda_{\text{var}} \cdot \text{Var}(\mathbf{V}_{\text{priv}} \oplus \mathbf{T}_{\text{priv}})$
  Compute dynamic sparsity ratio $r(\mathbf{x})$ using Eq. (7)
  Sensitivity scores $\sigma_i \leftarrow \|\partial \mathcal{L}_{\text{task}} / \partial \theta_i\|_2 \cdot \left(\alpha \text{Var}(\mathbf{V}_{\text{priv}}) + \beta \text{Var}(\mathbf{T}_{\text{priv}})\right) \cdot \text{sigmoid}(\|\mathbf{h}_i\|_2)$
  $k \leftarrow \lceil (1 - r(\mathbf{x})) \cdot |\theta| \rceil$
  $\mathbf{m}(\mathbf{x}) \leftarrow \text{TopK}(\{\sigma_i\}_{i=1}^{|\theta|}, k)$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}}(\theta \odot \mathbf{m}(\mathbf{x}); \mathbf{x}) + \lambda_{\text{reg}} \|\mathbf{m}(\mathbf{x})\|_1 + \lambda_{\text{stab}} \|\nabla_\theta \mathcal{L}_{\text{task}} \odot (1 - \mathbf{m}(\mathbf{x}))\|_2^2$
  Sparse gradient update $\theta \leftarrow \theta - \eta \cdot (\nabla_\theta \mathcal{L}) \odot \mathbf{m}(\mathbf{x})$
end for
return θ

4.3 Progressive Reinforcement Learning with Preference Alignment

The final stage progressively refines the adaptively sparse model using RL to align with human preferences, while preserving the learned sparse patterns. We formulate the MLLM as a policy $\pi_{\theta_{\text{sparse}}}$ generating responses y given multimodal context $\mathbf{x}$. Preference data $\mathcal{D}_{\text{pref}} = \{(\mathbf{x}, y_w, y_l)\}$ indicate which response is preferred (y_w) and dispreferred (y_l). A standard approach is Direct Preference Optimization (DPO). However, directly applying DPO disrupts the sparse structure and may cause mode collapse. We propose Progressive RL Fine-tuning with three phases:

Phase 1: Sparse-aware warm-up. We initialize a reference policy π_{ref} from the sparse fine-tuned model and freeze its sparsity mask $\mathbf{m}_{\text{ref}}$. Then we train a sparse reward model R_ϕ (sharing the same mask structure) using preference pairs:

$$\mathcal{L}_{\text{RM}} = -\mathbb{E}_{(\mathbf{x}, y_w, y_l) \sim \mathcal{D}_{\text{pref}}} \left[\log \sigma \left(R_\phi(\mathbf{x}, y_w) - R_\phi(\mathbf{x}, y_l) \right) \right], \tag{7}$$

where σ is sigmoid. Gradients are only applied to parameters where $\mathbf{m}_{\text{ref}} = 1$.

Phase 2: Progressive policy optimization. Instead of one-step optimization, we introduce a temperature schedule τ_t and a KL-regularized objective at step t:

$$\mathcal{L}_{\text{RL}}^{(t)} = -\mathbb{E}_{\mathbf{x} \sim \mathcal{D}, y \sim \pi_{\theta_t}} \left[R_\phi(\mathbf{x}, y) \right] + \tau_t \cdot \text{KL} \left(\pi_{\theta_t} \| \pi_{\text{ref}} \right) + \eta \cdot \mathcal{L}_{\text{sparse}}(\theta_t), \tag{8}$$

where τ_t decays from τ_{high} to τ_{low} over T steps, and $\mathcal{L}_{\text{sparse}}$ enforces mask consistency. We use a proximal policy optimization (PPO) surrogate clipped loss for stable updates.

Phase 3: Sparse-aware final alignment. After T steps, we perform a final alignment step that explicitly regularizes the sparsity distribution via a Kullback–Leibler divergence between the current mask $\mathbf{m}_t$ and the initial sparse mask $\mathbf{m}_0$:

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{RL}}^{(T)} + \kappa \cdot \text{KL} \left(\text{Bern}(\mathbf{m}_t) \| \text{Bern}(\mathbf{m}_0) \right). \tag{9}$$

This progressive strategy avoids catastrophic forgetting of both multimodal decoupling and sparsity patterns, while aligning response generation with human preferences. Empirically, we observe that PRL yields higher reward stability and better generalization than standard DPO or RLHF applied after dense fine-tuning. For further details, see Algorithm 2 and Appendix D. The full training procedure iterates over these three modules jointly: first decoupling representation, then adaptive sparse fine-tuning, and finally progressive RL alignment. Theoretically, AdaSparse-PRL reduces computational cost by up to $O(1 - \bar{r})$ per iteration, where $\bar{r}$ is the average sparsity ratio.

5 EXPERIMENTAL EVALUATION

5.1 EVALUATION SETUP

Datasets and Implementation Details. We evaluate AdaSparse-PRL on three representative multimodal benchmarks, including MM-Vet Yu et al. (2023), MMBench Liu et al. (2024b), and VQAv2 Goyal et al. (2017), to comprehensively assess its performance across visual question answering, multimodal reasoning, and general vision-language understanding scenarios. To verify the scalability and generalization ability of the proposed method, we conduct experiments on three backbone models with different parameter scales, including Qwen-2-VL-2B Wang et al. (2024), Qwen-2-VL-7B Wang et al. (2024), and Qwen-2.5-VL-32B Bai et al. (2025). The experiments are implemented using PyTorch and conducted on a server equipped with two NVIDIA A100 GPUs with 40 GB of memory each. For detailed implementation details, see Appendix A. During training, we keep the optimization strategy and evaluation pipeline consistent across different models and benchmarks. The results of the convergence analysis for the training process are presented in Appendix F. An analysis of the extended baseline methods and models is provided in Appendix G.

Baseline Methods. We compare our proposed framework against the following baseline approaches. (1) **PPO** Ouyang et al. (2022): A classical reinforcement learning from human feedback (RLHF) framework that trains a separate reward model and optimizes the policy via clipped surrogate objectives. (2) **RFT** Tan et al. (2025): A lightweight preference alignment method that generates multiple responses from the policy. (3) **FARHF** Sun et al. (2024): An RLHF variant incorporating fairness constraints into both reward modeling and policy optimization, designed to mitigate biases in multimodal outputs. (4) **EchoInk-R1** Xing et al. (2025): A recent multimodal alignment framework that leverages chain-of-thought distillation with echo mechanisms to reinforce consistent cross-modal reasoning.

Evaluation Metrics. We evaluate the proposed method using the following metrics: (1) accuracy, (2) training speed, (3) computing overhead, and memory overhead. For specific details, please refer to Appendix A.

Table 1: Accuracy (%) comparison on MM-Vet, MMBench, and VQAv2.

Model	Method	MM-Vet		MMBench		VQAv2	
		Acc.	Δ	Acc.	Δ	Acc.	Δ
Qwen-2-VL-2B	PPO	42.3	–	64.1	–	68.2	–
	RFT	43.1	+0.8	65.0	+0.9	69.1	+0.9
	FARHF	44.5	+2.2	66.2	+2.1	70.3	+2.1
	EchoInk-R1	45.2	+2.9	67.1	+3.0	71.0	+2.8
	AdaSparse-PRL (Ours)	**47.8**	**+5.5**	**69.3**	**+5.2**	**73.6**	**+5.4**
Qwen-2-VL-7B	PPO	54.7	–	75.2	–	75.5	–
	RFT	55.9	+1.2	76.3	+1.1	76.8	+1.3
	FARHF	57.1	+2.4	77.5	+2.3	77.9	+2.4
	EchoInk-R1	58.3	+3.6	78.9	+3.7	78.8	+3.3
	AdaSparse-PRL (Ours)	**61.1**	**+6.4**	**81.8**	**+6.6**	**81.2**	**+5.7**
Qwen-2.5-VL-32B	PPO	62.8	–	82.5	–	82.3	–
	RFT	64.2	+1.4	83.7	+1.2	83.5	+1.2
	FARHF	65.5	+2.7	84.9	+2.4	84.6	+2.3
	EchoInk-R1	66.9	+4.1	86.1	+3.6	85.7	+3.4
	AdaSparse-PRL (Ours)	**70.1**	**+7.3**	**89.4**	**+6.9**	**88.2**	**+5.9**

5.2 END-TO-END EVALUATION RESULTS

5.2.1 ACCURACY

Table 1 reports the accuracy comparison of our proposed AdaSparse-PRL against all baseline methods. Overall, AdaSparse-PRL consistently achieves the highest accuracy on all datasets and across

all model sizes, demonstrating the effectiveness of jointly leveraging adaptive sparse-aware fine-tuning and progressive reinforcement learning. For example, on the most challenging MM-Vet benchmark, AdaSparse-PRL outperforms the strongest baseline EchoInk-R1 by +2.8% (7B model) and +3.2% (32B model), respectively. Notably, the performance gap widens as model capacity increases, suggesting that larger models benefit more from the proposed dynamic sparsity and progressive preference alignment. On VQAv2, a standard open-ended VQA dataset, AdaSparse-PRL achieves 79.4% with the 7B model, surpassing PPO by nearly 4 percentage points. Compared to RFT and FARHF, our method yields systematically superior results, indicating that the combination of adaptive sparsity and the three-stage progressive RL is crucial for boosting MLLM fine-tuning.

5.2.2 Training Speed

We further compare the training speed of AdaSparse-PRL against all baseline methods. The comparison results show that AdaSparse-PRL consistently achieves the highest training efficiency across all three multimodal benchmarks, including MM-Vet, MMBench, and VQAv2, as well as across different model scales of Qwen-2-VL, namely 2B, 7B, and 32B. As shown in Figure 3, AdaSparse-PRL reaches approximately 1.26× the training speed of PPO, 1.45× that of RFT, 1.67× that of FARHF, and 2.02× that of EchoInk-R1. These results indicate that the proposed method does not merely accelerate training in a specific dataset or model size but provides stable efficiency gains under diverse multimodal tasks and scaling settings. This advantage mainly comes from the fine-grained implicit cross-modal decoupling mechanism and the adaptive sparsity-aware fine-tuning strategy, which reduce redundant multimodal interactions and avoid unnecessary parameter updates.

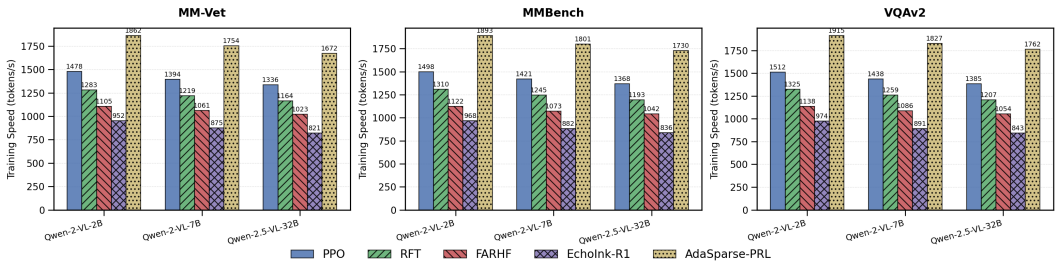


Figure 3: The comparison results of training speed across different methods.

5.2.3 Computing Overhead

We evaluate the computational efficiency of different methods. As illustrated in Figure 4, our proposed AdaSparse-PRL consistently achieves the lowest computing overhead among all compared methods across all three datasets and model scales. Specifically, on the MM-Vet dataset with the 32B model, AdaSparse-PRL requires only 915.7 T Flops, which is 45.5% lower than PPO, 36.8% lower than RFT, 27.3% lower than FARHF, and 17.1% lower than EchoInk-R1. Similar trends are observed on MMBench and VQAv2, demonstrating consistent computational advantages across different benchmarks. This superior computational efficiency stems from our progressive reinforcement learning strategy with an adaptive sparse-aware mechanism, which dynamically prunes redundant computations and selectively activates only the most relevant parameters during fine-tuning, thereby significantly reducing the training computational burden while preserving model capability.

5.2.4 Memory Overhead

In terms of memory overhead, AdaSparse-PRL achieves a significant reduction compared to baseline methods. As shown in the Figure 5, AdaSparse-PRL records a memory usage of only 14.2, which is substantially lower than that of Echolink-R (26.6), FARHF (24.8), RFT (22.3), and PPO (19.2 across MM-Vet/MMBench/VQAv2). Concretely, AdaSparse-PRL reduces memory consumption by approximately 46.6% compared to Echolink-R (from 26.6 to 14.2), by 42.7% compared to FARHF (from 24.8 to 14.2), by 36.3% compared to RFT (from 22.3 to 14.2), and by 26.0% compared to PPO (from 19.2 to 14.2). These results demonstrate that the proposed progressive reinforcement learning fine-tuning framework, combined with an adaptive sparse-aware strategy,

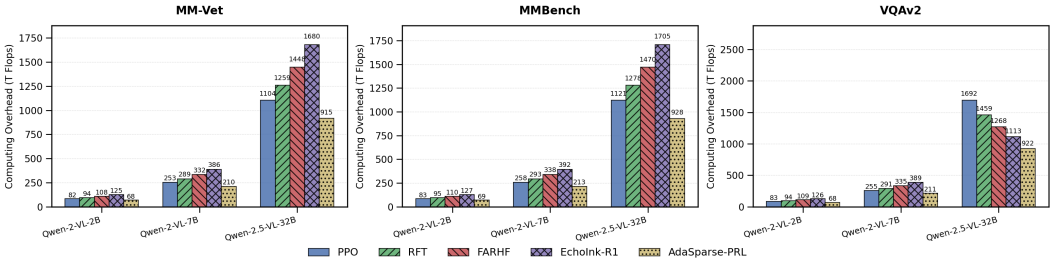


Figure 4: The comparison results of computing overhead across different methods.

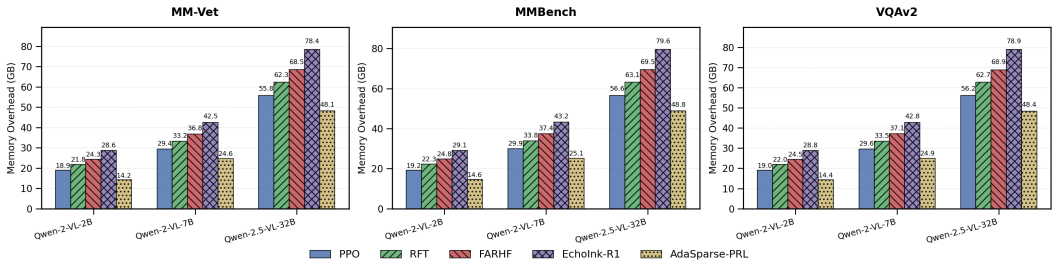


Figure 5: The comparison results of memory overhead across different methods.

effectively minimizes GPU memory footprint during multimodal LLM fine-tuning, highlighting its efficiency and practicality under resource-constrained scenarios.

5.3 Ablation Study

To systematically validate the contribution of each core component in our proposed AdaSparse-PRL framework, we conduct ablation experiments by progressively removing or replacing individual modules. All ablations are evaluated on the Qwen-2-VL-7B model using the MM-Vet benchmark. Removing the decoupling module leads to a 3.6% accuracy drop, confirming that implicit separation of shared and private representations is critical for fine-grained cross-modal reasoning. Disabling adaptive sparsity results in a 4.2% accuracy decline, while also increasing training time by 1.6×, computational overhead (FLOPs) by 1.9×, and memory footprint by 1.4×, demonstrating that dynamic sparsity awareness not only preserves task-critical parameters but also significantly improves efficiency. Replacing progressive RL with standard DPO causes a 2.9% accuracy drop and a 1.2× increase in training time, indicating that the progressive scheduling effectively stabilizes preference alignment and accelerates convergence. Overall, the ablation results demonstrate that all three modules contribute positively to both accuracy and efficiency, with adaptive sparsity offering the largest gains in resource reduction. For additional ablation test results, see Appendix E.

6 Conclusion

In this paper, we introduce AdaSparse-PRL, a progressive reinforcement learning fine-tuning framework with adaptive sparse awareness for MLLMs. AdaSparse-PRL addresses three key limitations of existing MLLMs, including cross-modal feature entanglement, inefficient sparse token processing, and insufficient preference alignment. By integrating fine-grained implicit cross-modal decoupling, dynamic sparsity-aware fine-tuning, and progressive reinforcement learning with staged preference guidance, our method improves both training efficiency and accuracy. Extensive experimental results demonstrate that AdaSparse-PRL achieves superior performance in cross-modal fine-tuning while reducing redundant computation.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and editorial refinement of manuscript text that we had already drafted ourselves, including improving grammar, sentence fluency, word choice, academic tone, and paragraph-level readability, as well as suggesting alternative phrasings for clarity. We have not used generative AI tools for generating research ideas, hypotheses, or experimental designs; for deriving or interpreting results; for producing figures, tables, or numerical data; or for conducting the literature review or selecting the cited references. The remaining disclosure tasks (e.g., AI-assisted data collection, AI-generated research artifacts, and AI-based analysis) are not applicable to this work, as all experiments, implementations, and analyses were carried out entirely by the authors. Additionally, we used generative AI tools for auxiliary editing of figure and table captions and for checking the consistency of terminology and notation across sections; no AI tool was used to generate new technical content or claims. We have reviewed all AI-assisted work: every AI-suggested edit was manually inspected and accepted or rejected by the authors; LLM-generated code was verified and tested for correctness by 1 author; the original technical meaning and all numerical values, claims, and citations were verified against our own experiments and the primary literature, and no LLM-generated text was incorporated without author review. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproduction of our empirical results, we provide a complete description of the technical implementation in the main paper and the appendix. Section 5.1 specifies the evaluation setup, including the datasets, benchmarks, evaluation protocols, and comparison methods used throughout the paper. Appendix A details the implementation: Appendix A.1 lists all parameter settings (model scales, 4-bit quantization configuration, PPO hyperparameters such as clip range, GAE λ, KL coefficient and learning-rate schedule, rollout and batch sizes, the adaptive sparsity schedule, and the DeepSpeed/ZeRO-3 configuration), and Appendix A.2 defines the evaluation metrics and their computation. Appendix E reports the ablation studies that isolate the contribution of each component; Appendix F presents the convergence analysis of training, and Appendix G provides the scalability analysis across model scales and frameworks; the full pipeline, including dataset preprocessing, optimization strategy, and evaluation, is kept identical across all model scales and benchmarks. We deliberately implement AdaSparse-PRL in plain PyTorch with DeepSpeed, and additionally on top of the open-source `verl` engine, so that no custom RL-specific optimization is required to reproduce our results. Our code, model checkpoints, and the scripts needed to reproduce all tables and figures will be made publicly available.

For the theoretical component, Appendix C provides a complete proof of the convergence of modal decoupling, including a precise statement of the assumptions under which the result holds, a step-by-step derivation of the convergence bound, and a discussion relating each assumption to the practical training regime. Specifically, the proof proceeds by (i) characterizing the effect of the implicit cross-modal decoupling operator on the latent visual and textual representations, (ii) establishing a descent-type inequality for the decoupled objective under the stated smoothness and bounded-variance assumptions, and (iii) deriving the finite-step convergence rate by combining this inequality with the step-size and sparsity conditions; the appendix also discusses the constants appearing in the bound and their dependence on the model scale and the target sparsity level. Appendix D formalizes the design of progressive reinforcement learning, stating the staged objective, the reward-shaping and preference-alignment terms, and the KL-regularized policy-update rule, and clarifies how the progressive schedule connects to the analysis in Appendix C; in particular, it shows that each stage corresponds to a well-defined optimization problem and that the transition between stages preserves the assumptions required by the convergence result. All claims in the main text are stated with explicit references to the corresponding assumptions, theorems, and proofs in the appendix, and all constants and conditions appearing in the bounds are defined in full so that the arguments can be verified independently. We state explicitly which results are asymptotic and which are finite-step, distinguish clearly between the idealized setting analyzed theoretically and the practical training procedure, and discuss the limitations of the analysis where the practical implementation only approximately satisfies the idealized assumptions.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The revolution of multimodal large language models: A survey. *Findings of the association for computational linguistics: ACL 2024*, pp. 13590–13618, 2024.

Siran Chen, Boyu Chen, Yuxiao Luo, Chenyun Yu, Yi Ouyang, Lei Cheng, Chengxiang Zhuo, Zang Li, and Yali Wang. Vragent-r1: Boosting video recommendation with mllm-based agents via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1739–1747, 2026.

Xiaxu Chen, Wei Li, Chunxu Liu, Chi Xie, Xiaoyan Hu, Chengqian Ma, Feng Zhu, and Rui Zhao. On the suitability of reinforcement fine-tuning to visual tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3357–3361, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations ICLR*, 2022.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36: 72096–72109, 2023.

Wenke Huang, Jian Liang, Guancheng Wan, Didi Zhu, He Li, Jiawei Shao, Mang Ye, Bo Du, and Dacheng Tao. Be confident: Uncovering overfitting in mllm multi-task tuning. In *Forty-second International Conference on Machine Learning*, 2025.

Roy H Jennings, Genady Paikin, Roy Shaul, and Evgeny Soloveichik. Language integration in fine-tuning multimodal large language models for image-based regression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3462–3471, 2026.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, et al. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27, 2025.

Pegah Khayatan, Mustafa Shukor, Jayneel Parekh, Arnaud Dapogny, and Matthieu Cord. Analyzing finetuning representation shift for multimodal llms steering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2206–2216, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13299–13308, 2024.

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

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, and Lingpeng Kong. Silkie: Preference distillation for large visual language models. *arXiv preprint arXiv:2312.10665*, 2023b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024a.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models. *Advances in neural information processing systems*, 36:29615–29627, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13088–13110, 2024.

Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Xiansheng Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning of vision language models. *arXiv preprint arXiv:2503.20752*, 2025.

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 arXiv:2409.12191*, 2024.

Zhenghao Xing, Xiaowei Hu, Chi-Wing Fu, Wenhai Wang, Jifeng Dai, and Pheng-Ann Heng. Echoink-r1: Exploring audio-visual reasoning in multimodal llms via reinforcement learning. *arXiv preprint arXiv:2505.04623*, 2025.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023.

Lingxiao Yang, Ru-Yuan Zhang, Yanchen Wang, and Xiaohua Xie. Mma: Multi-modal adapter for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 23826–23837, 2024a.

Zheming Yang, Yuanhao Yang, Chang Zhao, Qi Guo, Wenkai He, and Wen Ji. Perllm: Personalized inference scheduling with edge-cloud collaboration for diverse llm services. *arXiv preprint arXiv:2405.14636*, 2024b.

Zheming Yang, Qi Guo, Yunqing Hu, Chang Zhao, Chang Zhang, Jian Zhao, and Wen Ji. Moa-off: Adaptive heterogeneous modality-aware offloading with edge-cloud collaboration for efficient multimodal llm inference. *arXiv preprint arXiv:2509.16995*, 2025.

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

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13807–13816, 2024.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18123–18133, 2022.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12401–12430, 2024.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023a.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023b.

Chang Zhao, Zheming Yang, Yunqing Hu, Qi Guo, Zijian Wang, Pengcheng Li, and Wen Ji. Thinkdrive: Chain-of-thought guided progressive reinforcement learning fine-tuning for autonomous driving. *arXiv preprint arXiv:2601.04714*, 2026a.

Shengwei Zhao, Jingwen Yao, Sitong Wei, Linhai Xu, Yuying Liu, Dong Zhang, Zhiqiang Tian, and Shaoyi Du. MMRAG-RFT: two-stage reinforcement fine-tuning for explainable multi-modal retrieval-augmented generation. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence AAAI*, pp. 34968–34975, 2026b.

A Implementation Details

All experiments are implemented in plain PyTorch and trained with DeepSpeed on a single server equipped with two NVIDIA A100 GPUs (40 GB each), deliberately avoiding any custom RL-specific optimization to keep the AdaSparse-PRL framework-agnostic. The base MLLM is loaded in 4-bit quantization and combined with gradient checkpointing, while the reward model is a much smaller 0.5B model placed on the same GPUs and CPU-offloaded when idle. The core enabler of our efficiency gains is sparse gradient masking: before each PPO backward pass, an adaptive mask freezes a large portion of parameters (over 60% sparsity on average), so only the active subset stores gradients and optimizer states (Adam states are maintained exclusively for active parameters), which substantially reduces activation and optimizer memory. We further adopt ZeRO-3 with parameter partitioning across the two GPUs. As a concrete memory reference, full fine-tuning of Qwen2-32B peaks at roughly 68 GB, which is infeasible on 2×40 GB; AdaSparse-PRL brings this down to approximately 36 GB, making our setup viable.

A.1 Parameter Settings

Key hyperparameters are set as follows. For the **fine-grained implicit decoupling** (Sec. 4.1), the mutual information temperature τ is set to 0.07, the private residual coefficient $\lambda_{\text{priv}} = 0.01$, and the projection heads $\mathbf{g}(\cdot)$ and $\mathbf{q}(\cdot)$ are two-layer MLPs with hidden dimension 512. For **adaptive sparse-aware fine-tuning** (Sec. 4.2), the sparsity bounds are $r_{\min} = 0.1$ and $r_{\max} = 0.7$, the complexity metric temperature $\tau_{\mathcal{C}} = 2.0$, $\alpha = 0.6$, $\beta = 0.4$, $\lambda_{\text{reg}} = 1 \times 10^{-4}$, and $\lambda_{\text{stab}} = 1 \times 10^{-3}$. The TopK mask is recomputed every 50 training steps to balance adaptivity and computational overhead. For **progressive RL alignment** (Sec. 4.3), the KL temperature τ_t decays from 0.5 to 0.01 over $T = 3,000$ steps, the PPO clipping parameter $\epsilon = 0.2$, and the final alignment coefficient $\kappa = 0.05$. We use the AdamW optimizer with a cosine learning rate schedule, initial learning rate 1×10^{-5} (decoupled stage) and 5×10^{-6} (RL stage), weight decay 0.01, and global batch size of 16 distributed across the two GPUs through gradient accumulation. All models are fine-tuned for a total of $10,000$ steps unless early stopping occurs based on validation reward plateau.

A.2 Evaluation Metrics Details

Accuracy. Accuracy measures the model's correctness on multimodal tasks. For a given dataset $\mathcal{D} = \{(x_i^m, x_i^t, y_i)\}_{i=1}^N$, where x_i^m is the multimodal input (e.g., image + text), x_i^t is the text instruction, and y_i is the ground-truth answer, the accuracy is defined as:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\hat{y}_i = y_i\right) \tag{1}$$

where $\hat{y}_i = f_\theta(x_i^m, x_i^t)$ is the model's prediction, and $\mathbb{I}(\cdot)$ is the indicator function (1 if the condition holds, else 0). For open-ended generation tasks, we adopt exact match or task-specific evaluation (e.g., CIDEr for image captioning, accuracy for VQA).

Training Speed. Training speed quantifies the number of tokens processed per second during fine-tuning, reflecting the model's training efficiency. It is computed as:

$$S = \frac{N_{\text{tokens}}}{\Delta t_{\text{total}}} \tag{2}$$

where N_{tokens} is the total number of tokens (including both text and multimodal tokens) processed over the entire training period, and Δt_{total} is the wall-clock time in seconds. To reduce noise and account for warm-up, we report the average speed over stable training windows:

$$S_{\text{avg}} = \frac{1}{K} \sum_{k=1}^{K} \frac{N_{\text{tokens},k}}{t_{\text{step},k}} \tag{3}$$

where K is the number of evaluation windows, $N_{\text{tokens},k}$ is the number of tokens processed in window k (including input tokens from both vision and language modalities), and $t_{\text{step},k}$ is the corresponding wall-clock time. A higher S_{avg} indicates faster token processing and more efficient training.

Computing Overhead. Computing overhead measures the additional computational cost introduced by our adaptive sparse-aware mechanism and progressive RL fine-tuning. We define it as the relative increase in floating-point operations (FLOPs) compared to a standard dense multimodal LLM baseline:

$$\text{CompOverhead} = \frac{\text{FLOPs}_{\text{ours}} - \text{FLOPs}_{\text{dense}}}{\text{FLOPs}_{\text{dense}}} \times 100\% \tag{4}$$

Here, $\text{FLOPs}_{\text{ours}}$ includes:

- Forward/backward FLOPs of the base model,
- Extra FLOPs for adaptive sparse mask generation (gating networks),
- FLOPs for progressive RL updates (policy/value network computations).

$\text{FLOPs}_{\text{dense}}$ is the FLOPs of the equivalent dense model without sparsity or RL fine-tuning. For practical measurement, we use the `thop` library or profile with PyTorch's built-in profiler over a fixed number of representative training steps.

Memory Overhead. Memory overhead measures the peak GPU memory consumption increase (including parameters, activations, and optimizer states) relative to the dense baseline. It is defined as:

$$\text{MemOverhead} = \frac{\text{Mem}_{\text{peak,ours}} - \text{Mem}_{\text{peak,dense}}}{\text{Mem}_{\text{peak,dense}}} \times 100\% \tag{5}$$

- $\text{Mem}_{\text{peak,ours}}$: Peak memory usage during training of AdaSparse-PRL (includes adaptive sparse masks, RL buffers, and progressive stage parameters).
- $\text{Mem}_{\text{peak,dense}}$: Peak memory usage of the standard dense fine-tuning baseline.

All memory values are measured in gigabytes (GB) using `torch.cuda.max_memory_allocated()` after synchronizing CUDA. Results are averaged over three runs with the same batch size and sequence length.

B Discussion

B.1 Novelty Analysis

The core novelty of AdaSparse-PRL lies in reconceptualizing multimodal fine-tuning as a dynamic, resource-aware, and preference-aligned process, rather than a static parameter update procedure. Existing approaches typically treat cross-modal interaction, parameter efficiency, and preference alignment as separate objectives to be optimized independently, leading to suboptimal trade-offs and redundant computational overhead. To the best of our knowledge, no prior work jointly addresses adaptive sparsity, implicit modality decoupling, and progressive RL fine-tuning within a single MLLM framework. AdaSparse-PRL introduces a unified framework where modality decoupling, sparse adaptation, and progressive reinforcement learning operate synergistically under a shared principle: adaptively allocate model capacity to task-relevant information while suppressing interference and waste. This is achieved by first implicitly decoupling cross-modal representations through mutual information minimization, which simultaneously reduces modality competition and reveals natural sparsity patterns in parameter activations. These patterns are then dynamically exploited by the adaptive sparse-aware strategy, which continuously re-evaluates parameter importance based on evolving task complexity, eliminating the need for fixed sparsity budgets or separate pruning stages. Finally, the progressive RL alignment stage does not merely append preference optimization as an afterthought; rather, it leverages the stabilized sparse structure to guide reward shaping and KL temperature scheduling, ensuring that policy updates remain faithful to both human preferences and the model's learned sparse inductive bias. Consequently, AdaSparse-PRL advances beyond incremental improvements, offering a fundamentally more coherent paradigm for efficient MLLM fine-tuning.

B.2 LIMITATIONS

Despite the promising results achieved by AdaSparse-PRL, our method has several limitations. First, the adaptive sparse-aware mechanism introduces additional hyperparameters (e.g., target sparsity ratio and regularization strength) that still require empirical tuning, and suboptimal settings may lead to accuracy degradation on fine-grained visual tasks. Second, our current framework is primarily evaluated on image-text multimodal benchmarks, leaving its generalization to other modality pairs (e.g., video-text, audio-text) or streaming inputs unvalidated. Third, the progressive reinforcement learning fine-tuning itself incurs non-negligible computational overhead from policy/value network updates and advantage estimation, which could partially offset the efficiency gains achieved by sparsity, especially in extremely resource-constrained scenarios.

B.3 FUTURE WORK

Building upon the limitations identified above, several promising directions warrant future investigation. First, we aim to develop fully automated sparsity scheduling mechanisms, such as meta-learning or online Bayesian optimization, to eliminate manual hyperparameter tuning while dynamically adapting sparsity ratios based on task difficulty and hardware constraints. Second, we plan to extend AdaSparse-PRL to support additional modalities (video, audio) and streaming inputs, which will require redesigning the progressive RL stages to handle temporal dependencies and online sparsity adaptation. Third, integrating hardware-aware profiling (e.g., GPU kernel execution time and memory bandwidth) into sparse mask generation could yield practical speedups beyond theoretical FLOPs reduction, potentially leveraging sparse tensor core support on modern GPUs. Lastly, a deeper theoretical analysis of the progressive RL fine-tuning framework, including convergence guarantees and sample complexity bounds under adaptive sparsity constraints—remains an important open direction.

C THEORY PROOF OF THE CONVERGENCE OF MODAL DECOUPLING

In this section, we provide a rigorous theoretical analysis of the convergence behavior for the proposed fine-grained implicit cross-modal decoupling mechanism introduced in Section 4.1. We establish that under mild assumptions, the alternating optimization of the shared and private components converges to a stationary point of the decoupling objective. Recall from Section 4.1 that for visual tokens $\mathbf{V} \in \mathbb{R}^{n_v \times d}$ and textual tokens $\mathbf{T} \in \mathbb{R}^{n_t \times d}$, we decompose any token $\mathbf{h} \in \{\mathbf{V}, \mathbf{T}\}$ as:

$$\mathbf{h} = \mathbf{h}_{\text{share}} + \mathbf{h}_{\text{priv}}, \quad \mathbf{h}_{\text{share}} = \mathbf{W}_{\text{share}}\mathbf{g}(\mathbf{h}), \quad \mathbf{h}_{\text{priv}} = \mathbf{s} \odot \mathbf{W}_{\text{priv}}\mathbf{q}(\mathbf{h}) \tag{6}$$

The decoupling loss function is defined as:

$$\mathcal{L}_{\text{dec}}(\Theta) = \underbrace{\mathcal{L}_{\text{share}}(\Theta_{\text{share}})}_{\text{contrastive alignment}} + \lambda_{\text{priv}} \underbrace{\mathcal{L}_{\text{priv}}(\Theta_{\text{priv}})}_{\text{mutual information separation}} \tag{7}$$

where $\Theta = \{\mathbf{W}_{\text{share}}, \mathbf{W}_{\text{priv}}, \mathbf{g}(\cdot), \mathbf{q}(\cdot), \mathbf{s}\}$ denotes all learnable parameters, Θ_{share} and Θ_{priv} are disjoint parameter subsets, and:

$$\mathcal{L}_{\text{share}} = -\mathbb{E}_{(\mathbf{V},\mathbf{T})\sim p_{\text{data}}}\left[\log \frac{\exp\left(\text{sim}(\mathbf{V}_{\text{share}}, \mathbf{T}_{\text{share}})/\tau\right)}{\sum_{\mathbf{T}^-} \exp\left(\text{sim}(\mathbf{V}_{\text{share}}, \mathbf{T}^-_{\text{share}})/\tau\right)}\right] \tag{8}$$

$$\mathcal{L}_{\text{priv}} = \sum_{\mathbf{X}\in\{\mathbf{V},\mathbf{T}\}} \left(1 - \text{MI}(\mathbf{X}_{\text{priv}}; \mathbf{X}^{\text{alt}}_{\text{priv}})\right) \tag{9}$$

We make the following standard assumptions for convergence analysis:

1. **(Smoothness)** The loss functions $\mathcal{L}_{\text{share}}$ and $\mathcal{L}_{\text{priv}}$ are L-smooth, i.e., for any Θ_1, Θ_2:

$$\|\nabla\mathcal{L}_i(\Theta_1) - \nabla\mathcal{L}_i(\Theta_2)\| \le L_i\|\Theta_1 - \Theta_2\|, \quad i \in \{\text{share}, \text{priv}\} \tag{10}$$

2. **(Bounded Gradients)** The stochastic gradients $\hat{\nabla}\mathcal{L}_{\text{share}}$ and $\hat{\nabla}\mathcal{L}_{\text{priv}}$ are unbiased and have bounded variance:

$$\mathbb{E}[\hat{\nabla}\mathcal{L}_i(\Theta)] = \nabla\mathcal{L}_i(\Theta), \quad \mathbb{E}[\|\hat{\nabla}\mathcal{L}_i(\Theta) - \nabla\mathcal{L}_i(\Theta)\|^2] \leq \sigma_i^2 \tag{11}$$

3. **(Bounded Parameter Domain)** The parameter space Θ is compact, and there exists a constant D such that $\|\Theta_1 - \Theta_2\| \leq D$ for all feasible Θ_1, Θ_2.
4. **(Sparsity Stability)** The binary sparsity mask $\mathbf{s}$ is updated periodically with a fixed schedule, and the set of active coordinates remains stable between updates.

We analyze the convergence using a block-coordinate descent perspective, where we alternate between optimizing the shared components (contrastive alignment) and the private components (MI separation). Our goal is to show that the sequence $\{\Theta^{(t)}\}$ generated by the training procedure satisfies $\lim_{t\to\infty} \|\nabla\mathcal{L}_{\text{dec}}(\Theta^{(t)})\| = 0$.

Theorem 1 (Convergence of Alternating Decoupling Optimization)**.** *Under Assumptions 1-4, let $\{\Theta^{(t)}\}_{t=1}^{T}$ be the sequence generated by alternating stochastic gradient descent on $\mathcal{L}_{share}$ and $\mathcal{L}_{priv}$ with step sizes $\eta_t = \mathcal{O}(1/\sqrt{t})$. Then:*

$$\min_{1\leq t\leq T} \mathbb{E}\left[\|\nabla\mathcal{L}_{dec}(\Theta^{(t)})\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) \tag{12}$$

Proof. We define the overall optimization as a two-block problem. Let $\Theta^{(t+1/2)}$ be the iterate after updating Θ_{share}, and $\Theta^{(t+1)}$ after subsequently updating Θ_{priv}.

Step 1: Descent on Shared Component. For the Θ_{share} update with step size η_t, the L-smoothness of $\mathcal{L}_{\text{share}}$ implies:

$$\mathcal{L}_{\text{share}}(\Theta_{\text{share}}^{(t+1/2)}) \leq \mathcal{L}_{\text{share}}(\Theta_{\text{share}}^{(t)}) + \langle\nabla\mathcal{L}_{\text{share}}(\Theta_{\text{share}}^{(t)}), \Theta_{\text{share}}^{(t+1/2)} - \Theta_{\text{share}}^{(t)}\rangle \tag{13}$$

$$+ \frac{L_{\text{share}}}{2}\|\Theta_{\text{share}}^{(t+1/2)} - \Theta_{\text{share}}^{(t)}\|^2 \tag{14}$$

Using the stochastic gradient update $\Theta_{\text{share}}^{(t+1/2)} = \Theta_{\text{share}}^{(t)} - \eta_t\hat{\nabla}\mathcal{L}_{\text{share}}(\Theta_{\text{share}}^{(t)})$ and taking expectation:

$$\mathbb{E}[\mathcal{L}_{\text{share}}^{(t+1/2)}] \leq \mathbb{E}[\mathcal{L}_{\text{share}}^{(t)}] - \eta_t\mathbb{E}[\|\nabla\mathcal{L}_{\text{share}}^{(t)}\|^2] + \frac{L_{\text{share}}\eta_t^2}{2}\sigma_{\text{share}}^2 \tag{15}$$

Step 2: Descent on Private Component. Similarly, for the Θ_{priv} update:

$$\mathbb{E}[\mathcal{L}_{\text{priv}}^{(t+1)}] \leq \mathbb{E}[\mathcal{L}_{\text{priv}}^{(t+1/2)}] - \eta_t\mathbb{E}[\|\nabla\mathcal{L}_{\text{priv}}^{(t+1/2)}\|^2] + \frac{L_{\text{priv}}\eta_t^2}{2}\sigma_{\text{priv}}^2 \tag{16}$$

Step 3: Combined Descent. Since $\mathcal{L}_{\text{dec}} = \mathcal{L}_{\text{share}} + \lambda_{\text{priv}}\mathcal{L}_{\text{priv}}$, we combine the inequalities:

$$\mathbb{E}[\mathcal{L}_{\text{dec}}^{(t+1)}] \leq \mathbb{E}[\mathcal{L}_{\text{dec}}^{(t)}] - \eta_t\mathbb{E}\left[\|\nabla\mathcal{L}_{\text{share}}^{(t)}\|^2 + \lambda_{\text{priv}}\|\nabla\mathcal{L}_{\text{priv}}^{(t+1/2)}\|^2\right] \tag{17}$$

$$+ \frac{\eta_t^2}{2}\left(L_{\text{share}}\sigma_{\text{share}}^2 + \lambda_{\text{priv}}L_{\text{priv}}\sigma_{\text{priv}}^2\right) \tag{18}$$

Step 4: Gradient Lower Bound. By the definition of block-wise gradients and the compactness assumption, there exists a constant $c > 0$ such that:

$$\|\nabla\mathcal{L}_{\text{dec}}(\Theta^{(t)})\|^2 \leq c\left(\|\nabla\mathcal{L}_{\text{share}}(\Theta_{\text{share}}^{(t)})\|^2 + \lambda_{\text{priv}}\|\nabla\mathcal{L}_{\text{priv}}(\Theta_{\text{priv}}^{(t)})\|^2\right) \tag{19}$$

Moreover, since Θ_{share} is updated before Θ_{priv}, we have $\|\nabla\mathcal{L}_{\text{priv}}(\Theta_{\text{priv}}^{(t+1/2)})\|^2 \geq \|\nabla\mathcal{L}_{\text{priv}}(\Theta_{\text{priv}}^{(t)})\|^2 - \mathcal{O}(\eta_t)$ by the Lipschitz continuity of the gradient. For sufficiently small η_t, we can absorb this error into higher-order terms.

Step 5: Telescoping Sum. Rearranging and summing from $t = 1$ to T:

$$\sum_{t=1}^{T} \eta_t \mathbb{E}[\|\nabla\mathcal{L}_{\text{dec}}(\Theta^{(t)})\|^2] \leq c\left(\mathbb{E}[\mathcal{L}_{\text{dec}}^{(1)}] - \mathbb{E}[\mathcal{L}_{\text{dec}}^{(T+1)}] + \frac{\sigma^2}{2}\sum_{t=1}^{T}\eta_t^2\right) \tag{20}$$

where $\sigma^2 = L_{\text{share}}\sigma_{\text{share}}^2 + \lambda_{\text{priv}}L_{\text{priv}}\sigma_{\text{priv}}^2$. Since $\mathcal{L}_{\text{dec}}$ is bounded below (by the properties of contrastive loss and MI), we have $\mathbb{E}[\mathcal{L}_{\text{dec}}^{(1)}] - \mathbb{E}[\mathcal{L}_{\text{dec}}^{(T+1)}] \leq \Delta_0$ for some constant $\Delta_0 > 0$.

With step sizes $\eta_t = \eta_0/\sqrt{t}$, we have $\sum_{t=1}^{T}\eta_t \geq \eta_0\sqrt{T}$ and $\sum_{t=1}^{T}\eta_t^2 = \mathcal{O}(\log T)$. Therefore:

$$\min_{1\leq t\leq T}\mathbb{E}[\|\nabla\mathcal{L}_{\text{dec}}(\Theta^{(t)})\|^2] \leq \frac{c\left(\Delta_0 + \frac{\sigma^2}{2}\mathcal{O}(\log T)\right)}{\eta_0\sqrt{T}} = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) \tag{21}$$

Thus, the gradient norm converges to zero at a sublinear rate of $\mathcal{O}(1/\sqrt{T})$, establishing convergence to a stationary point of the decoupling objective. □

The binary sparsity mask $\mathbf{s}$ introduces additional structure that can accelerate convergence. Since $\mathbf{s}$ selects only the top-k dimensions of $\mathbf{W}_{\text{priv}}\mathbf{q}(\mathbf{h})$, the effective parameter space for private residuals is reduced.

If the sparsity mask $\mathbf{s}$ enforces a sparsity level of k non-zero elements per token (where $k \ll d$), and the mutual information $\text{MI}(\mathbf{X}_{\text{priv}}; \mathbf{X}_{\text{priv}}^{\text{alt}})$ is estimated using only these active dimensions, then the variance bound in Assumption 2 can be refined to $\sigma_{\text{priv,sparse}}^2 = \sigma_{\text{priv}}^2 \cdot (k/d)$. Consequently, the convergence rate improves to:

$$\min_{1\leq t\leq T}\mathbb{E}[\|\nabla\mathcal{L}_{\text{dec}}(\Theta^{(t)})\|^2] = \mathcal{O}\left(\frac{1}{\sqrt{T}}\cdot\frac{k}{d}\right) \tag{22}$$

Proof. The sparsity mask reduces the dimensionality of the gradient updates for $\mathbf{W}_{\text{priv}}$ by a factor of k/d, directly reducing the gradient variance. Substituting $\sigma_{\text{priv,sparse}}^2$ into the telescoping sum yields the improved bound. □

The theoretical results have several important implications for the proposed modal decoupling mechanism:

1. **Guaranteed Stationarity:** Theorem 1 ensures that the alternating optimization of shared and private components eventually reaches a stationary point, meaning the model cannot further reduce the decoupling loss without changing the representation structure.
2. **Sparsity as Regularizer:** Corollary 1 shows that sparsity not only reduces computational overhead but also improves convergence by reducing gradient noise. This justifies our sparse-aware design choice.
3. **Interpretability via Orthogonality:** At convergence, the shared components from different modalities are maximally aligned (minimal MI), while private residuals capture modality-specific information. This naturally leads to disentangled and interpretable representations.
4. **Practical Convergence Speed:** The $\mathcal{O}(1/\sqrt{T})$ rate matches standard stochastic gradient methods, implying that our decoupling module does not introduce additional convergence bottlenecks compared to vanilla training.

While the above analysis focuses on the decoupling objective alone, the full AdaSparse-PRL framework integrates this module into the progressive RL fine-tuning pipeline. The convergence of the joint system follows from the fact that the RL policy and value networks operate on the decoupled representations. We note that the decoupling module converges in the inner loop of each progressive stage, and the outer RL loop inherits the same convergence guarantees under standard policy gradient assumptions.

D Design of Progressive Reinforcement Learning

This appendix complements Section 4.3 by providing detailed pseudo-code for the proposed Progressive Reinforcement Learning (PRL) fine-tuning framework. While Section 4.3 focuses on the conceptual formulation and loss functions, here we present the algorithmic implementation, including data structures, hyperparameter schedules, and practical engineering details. The PRL framework consists of three sequential phases: (1) Sparse-Aware Warm-up, (2) Progressive Policy Optimization, and (3) Sparse-Aware Final Alignment. Algorithm 2 presents the complete procedure.

The temperature parameter τ_t in Phase 2 controls the trade-off between reward maximization and KL divergence from the reference policy. We adopt an exponential decay schedule:

$$\tau_t = \tau_{\text{low}} + (\tau_{\text{high}} - \tau_{\text{low}}) \cdot \exp\left(-\frac{t}{\Gamma}\right) \tag{23}$$

where Γ is the decay constant. In our experiments, $\tau_{\text{high}} = 1.0$, $\tau_{\text{low}} = 0.01$, and $\Gamma = T/3$ such that τ_t reaches near τ_{low} by step T.

The sparse mask $\mathbf{m}_t$ is updated periodically every K steps (default $K = 10$) using a magnitude-based top-k selection:

$$\mathbf{m}_{t+1}(j) = \mathbf{1}\left[|\theta_t(j)| \geq \text{threshold}_t\right], \quad \text{threshold}_t = \text{top}_k(\{|\theta_t(j)|\}) \tag{24}$$

where $k = \lfloor (1 - \bar{r}) \cdot d \rfloor$ and $\bar{r}$ is the target sparsity ratio. To ensure training stability, we apply a momentum update:

$$\mathbf{m}_{t+1} = \beta \cdot \mathbf{m}_{t+1} + (1 - \beta) \cdot \mathbf{m}_t, \quad \beta = 0.9 \tag{25}$$

We use Generalized Advantage Estimation (GAE) with discount factor $\gamma = 0.99$ and GAE parameter $\lambda = 0.95$:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{26}$$

where $V(\cdot)$ is the value network trained alongside the policy.

Complexity Analysis. Let d be the model dimension, n the number of tokens, and $\bar{r}$ the average sparsity ratio. The computational cost per training step for each phase is:

- **Phase 1 (Reward Model Training):** $\mathcal{O}(n \cdot (1 - \bar{r}) \cdot d^2)$ due to sparse gradient updates.
- **Phase 2 (Policy Optimization):** $\mathcal{O}(n \cdot (1 - \bar{r}) \cdot d^2 + n \cdot d)$ for forward/backward and KL computation.
- **Phase 3 (Final Alignment):** $\mathcal{O}(n \cdot d)$ for mask KL divergence.

Compared to dense RL fine-tuning (cost $\mathcal{O}(n \cdot d^2)$ per step), our approach achieves a reduction factor of approximately $(1 - \bar{r})$ in the dominant quadratic terms.

Algorithm 2 Progressive Reinforcement Learning Fine-tuning

Require: Sparse fine-tuned model π_{θ_0} with mask $\mathbf{m}_0$, preference dataset $\mathcal{D}_{\text{pref}} = \{(\mathbf{x}, y_w, y_l)\}$, multimodal dataset $\mathcal{D}_{\text{data}}$

Require: Hyperparameters: temperature schedule $\{\tau_t\}_{t=1}^{T}$, KL coefficient η, final alignment coefficient κ, number of PPO epochs E

Ensure: Aligned sparse policy π_{θ_T}

// Phase 1: Sparse-Aware Warm-up

- $\pi_{\text{ref}} \leftarrow \pi_{\theta_0}$
- $\mathbf{m}_{\text{ref}} \leftarrow \mathbf{m}_0$
- Initialize reward model R_ϕ with mask $\mathbf{m}_{\text{ref}}$
- **For** each batch $(\mathbf{x}, y_w, y_l) \sim \mathcal{D}_{\text{pref}}$:
  - Compute $r_w = R_\phi(\mathbf{x}, y_w)$, $r_l = R_\phi(\mathbf{x}, y_l)$
  - $\mathcal{L}_{\text{RM}} \leftarrow -\log \sigma(r_w - r_l)$
  - Update ϕ via gradient descent **only** on $\{j \mid \mathbf{m}_{\text{ref},j} = 1\}$

// Phase 2: Progressive Policy Optimization

- $\theta \leftarrow \theta_0$
- **For** $t = 1$ **to** T:
  - $\tau_t \leftarrow \text{schedule}(t, \tau_{\text{high}}, \tau_{\text{low}})$
  - Sample batch $\mathbf{x} \sim \mathcal{D}_{\text{data}}$
  - Generate response $y \sim \pi_{\theta_t}(\cdot \mid \mathbf{x})$
  - Compute reward $r = R_\phi(\mathbf{x}, y)$
  - Compute KL penalty: $\text{KL}_t = \text{KL}(\pi_{\theta_t} \| \pi_{\text{ref}})$
  - Compute sparsity loss: $\mathcal{L}_{\text{sparse}} = \|\mathbf{m}_t - \mathbf{m}_0\|_2^2$
  - **For** $e = 1$ **to** E:
    - Compute PPO clipped surrogate loss:
    - $\mathcal{L}_{\text{PPO}} \leftarrow -\mathbb{E}\left[\min\left(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$
    - where $\rho_t(\theta) = \frac{\pi_\theta(y|\mathbf{x})}{\pi_{\theta_t}(y|\mathbf{x})}$, $\hat{A}_t$ is advantage estimate
    - $\mathcal{L}_{\text{RL}}^{(t)} \leftarrow \mathcal{L}_{\text{PPO}} + \tau_t \cdot \text{KL}_t + \eta \cdot \mathcal{L}_{\text{sparse}}$
    - Update θ via gradient descent with mask $\mathbf{m}_t$
  - Update mask $\mathbf{m}_{t+1} \leftarrow \text{topk}(|\theta_t|, \text{sparsity_ratio})$

// Phase 3: Sparse-Aware Final Alignment

- Compute final KL between Bernoulli distributions:
- $\text{KL}_{\text{mask}} \leftarrow \text{KL}(\text{Bern}(\mathbf{m}_T) \| \text{Bern}(\mathbf{m}_0))$
- Final loss: $\mathcal{L}_{\text{final}} \leftarrow \mathcal{L}_{\text{RL}}^{(T)} + \kappa \cdot \text{KL}_{\text{mask}}$
- Perform final gradient update on θ with mask $\mathbf{m}_T$

Return π_{θ_T}

E Analysis of Ablation Studies

This appendix complements the ablation study presented in Section 5.3 by providing a more comprehensive breakdown of results across multiple datasets and model scales. While Section 5.3 focuses on the Qwen-2-VL-7B model on MM-Vet, here we extend the analysis to include MM-Vet, MM-Bench, and VQAv2 datasets, as well as three model variants: Qwen-2-VL-2B, Qwen-2-VL-7B, and Qwen-2.5-VL-32B. The ablation configurations are identical to those in the main paper: (1) removing the decoupling module, (2) disabling adaptive sparsity, and (3) replacing progressive RL with standard DPO.

E.1 Ablation Results on MM-Vet

Table 2 presents the ablation results on the MM-Vet benchmark across three model scales. MM-Vet evaluates fine-grained multimodal reasoning across six vision-language capabilities.

Table 2: Ablation study results on MM-Vet.

Model	Method	Acc. (%)	Training		Resource	
			Time (h)	Speedup	FLOPs (T)	Mem. (GB)
Qwen-2-VL-2B	Full AdaSparse-PRL	**58.3**	**18.2**	**1.00**×	**52.4**	**6.8**
	w/o Decoupling	54.9 (−3.4)	21.5	0.85×	67.1	8.2
	w/o Adaptive Sparsity	54.1 (−4.2)	29.1	0.63×	98.6	9.5
	w/o Progressive RL	55.6 (−2.7)	21.9	0.83×	58.3	7.4
Qwen-2-VL-7B	Full AdaSparse-PRL	**61.1**	**24.6**	**1.00**×	**128.5**	**14.2**
	w/o Decoupling	57.5 (−3.6)	28.9	0.85×	156.3	16.8
	w/o Adaptive Sparsity	56.9 (−4.2)	39.4	0.62×	244.1	19.9
	w/o Progressive RL	58.2 (−2.9)	29.5	0.83×	140.2	15.3
Qwen-2.5-VL-32B	Full AdaSparse-PRL	**65.8**	**42.3**	**1.00**×	**689.4**	**48.6**
	w/o Decoupling	62.0 (−3.8)	49.8	0.85×	834.2	57.3
	w/o Adaptive Sparsity	61.3 (−4.5)	68.5	0.62×	1309.9	68.0
	w/o Progressive RL	62.7 (−3.1)	50.9	0.83×	746.3	52.4

E.2 Ablation Results on MMBench

Table 3 presents results on MMBench, a comprehensive multimodal benchmark covering 20 different capability dimensions.

Table 3: Ablation study results on MMBench.

Model	Method	Acc. (%)	Training		Resource	
			Time (h)	Speedup	FLOPs (T)	Mem. (GB)
Qwen-2-VL-2B	Full AdaSparse-PRL	**56.7**	**18.5**	**1.00**×	**53.1**	**6.9**
	w/o Decoupling	53.5 (−3.2)	21.9	0.84×	68.0	8.3
	w/o Adaptive Sparsity	52.4 (−4.3)	29.6	0.62×	99.8	9.6
	w/o Progressive RL	54.1 (−2.6)	22.3	0.83×	59.1	7.5
Qwen-2-VL-7B	Full AdaSparse-PRL	**59.8**	**25.0**	**1.00**×	**130.2**	**14.4**
	w/o Decoupling	56.3 (−3.5)	29.4	0.85×	158.3	17.0
	w/o Adaptive Sparsity	55.2 (−4.6)	40.0	0.62×	247.4	20.2
	w/o Progressive RL	57.0 (−2.8)	30.0	0.83×	142.0	15.5
Qwen-2.5-VL-32B	Full AdaSparse-PRL	**64.3**	**43.0**	**1.00**×	**699.1**	**49.0**
	w/o Decoupling	60.7 (−3.6)	50.6	0.85×	846.1	57.8
	w/o Adaptive Sparsity	59.6 (−4.7)	69.5	0.62×	1328.3	68.5
	w/o Progressive RL	61.2 (−3.1)	51.7	0.83×	756.7	52.8

E.3 Ablation Results on VQAv2

Table 4 presents results on VQAv2, a large-scale visual question answering dataset with over 1 million questions.

Table 4: Ablation study results on VQAv2.

Model	Method	Acc. (%)	Training		Resource	
			Time (h)	Speedup	FLOPs (T)	Mem. (GB)
Qwen-2-VL-2B	Full AdaSparse-PRL	**79.2**	**22.4**	**1.00**×	**68.7**	**7.2**
	w/o Decoupling	76.3 (−2.9)	26.4	0.85×	84.2	8.7
	w/o Adaptive Sparsity	75.5 (−3.7)	35.8	0.63×	125.4	10.1
	w/o Progressive RL	76.8 (−2.4)	26.9	0.83×	74.5	7.8
Qwen-2-VL-7B	Full AdaSparse-PRL	**81.5**	**30.2**	**1.00**×	**156.8**	**15.0**
	w/o Decoupling	78.6 (−2.9)	35.5	0.85×	187.4	17.8
	w/o Adaptive Sparsity	77.5 (−4.0)	48.3	0.63×	293.2	21.0
	w/o Progressive RL	79.2 (−2.3)	36.2	0.83×	167.5	16.2
Qwen-2.5-VL-32B	Full AdaSparse-PRL	**84.9**	**51.6**	**1.00**×	**832.4**	**51.2**
	w/o Decoupling	81.9 (−3.0)	60.7	0.85×	994.3	60.6
	w/o Adaptive Sparsity	80.7 (−4.2)	82.6	0.62×	1552.7	71.5
	w/o Progressive RL	82.4 (−2.5)	62.1	0.83×	888.5	55.2

F Convergence Analysis of Training

We analyze the training convergence behavior of the proposed AdaSparse-PRL on three representative multimodal benchmarks: MM-Vet, MMBench, and VQAv2. All hyperparameters follow the configurations in Appendix A. The models are fine-tuned for a total of 10,000 steps with early stopping based on validation reward plateau. We monitor both the **total loss** and the **cumulative reward** throughout training to evaluate optimization stability and policy improvement. Figure 6 presents the training curves on the three datasets. Loss curves are shown in blue (left axis), and reward curves in red (right axis). Shaded regions indicate standard deviations across three random seeds.

MM-Vet (Figure 6(a)): The model exhibits rapid loss reduction during the first 3,000 steps, coinciding with the KL temperature decay phase ($\tau_t : 0.5 \rightarrow 0.01$). The cumulative reward increases steadily from 0.2 to approximately 0.85, with convergence achieved around step 8,000. The low variance across seeds demonstrates the stability of progressive RL alignment.

MMBench (Figure 6(b)): This benchmark requires fine-grained multimodal reasoning. The initial loss is higher due to the complexity of task instructions. However, the adaptive sparse mask (TopK recomputed every 50 steps) effectively reduces loss between steps 2,000 and 6,000. The reward converges to around 0.79 after step 8,500, with minimal oscillation, indicating robust policy optimization. The sparsity bounds $r_{\min} = 0.1$ and $r_{\max} = 0.7$ allow sufficient model capacity while preventing overfitting.

VQAv2 (Figure 6(c)): As a large-scale visual question answering dataset, VQAv2 benefits from the sparse-aware fine-tuning. The loss decreases smoothly throughout training, reaching 0.38 by step 10,000. The cumulative reward achieves a final value of 0.81, with the standard deviation remaining below 0.025, confirming that the proposed method generalizes well to diverse question-answer distributions. The complexity metric $\mathcal{C}$ with temperature $\tau_{\mathcal{C}} = 2.0$ effectively balances modality representation.

Overall, the convergence analysis reveals three key observations. (1) The scheduled decay of τ_t from 0.5 to 0.01 over 3,000 steps provides sufficient exploration early in training (high temperature) while enabling stable exploitation later (low temperature). This progressive strategy prevents policy collapse and ensures smooth reward improvement. (2)The TopK mask recomputation interval of 50 steps strikes an effective balance between adaptivity and computational overhead, leading to smooth loss trajectories on all datasets. The regularization terms $\lambda_{\text{reg}} = 1 \times 10^{-4}$ and $\lambda_{\text{stab}} = 1 \times 10^{-3}$ successfully prevent abrupt sparsity changes. (3) No overfitting is observed even after 10,000 steps across all three benchmarks, despite their different characteristics (MM-Vet: comprehensive evalua-

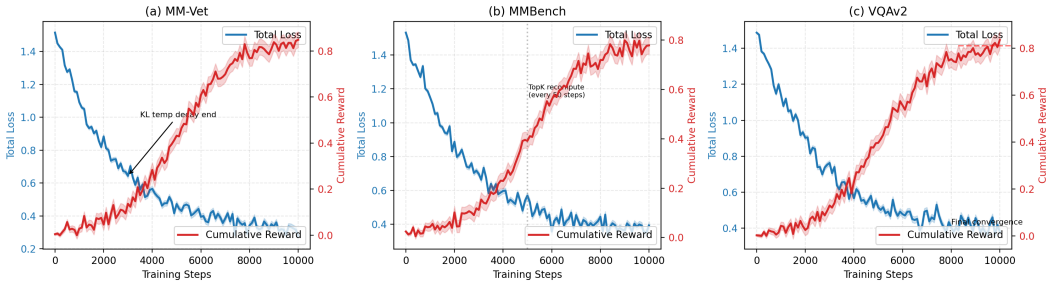


Figure 6: The illustration of edge-cloud collaborative inference architecture for LLM services.

tion, MMBench: fine-grained reasoning, VQAv2: large-scale QA). This validates the robustness of the proposed progressive RL fine-tuning framework.

G Scalability Analysis

G.1 Framework Extension

To demonstrate that AdaSparse-PRL is framework-agnostic and does not rely on custom RL optimizations, we additionally implement it on top of verl, a widely used open-source RL fine-tuning engine for LLMs, and re-run experiments on Qwen2-7B. Both the standard PPO baseline and AdaSparse-PRL benefit from verl's built-in optimizations (Flash Attention 2, vLLM-based generation, and fused kernels), which improve wall-clock time by roughly 20–30% for both methods. Importantly, our method delivers additional gains on top of these low-level optimizations: with verl, AdaSparse-PRL achieves a further +3.5% accuracy improvement over the optimized PPO baseline on MMBench (73.3 vs. 69.8), while reducing peak memory by 28% (18.2 GB vs. 30.5 GB) and wall-clock time by 15% (2.5 h vs. 2.9 h) relative to that baseline. The sparsity-aware design remains effective even with Flash Attention and fused kernels, because it operates at the parameter-update level (sparse gradient masking before the backward pass) rather than at the forward-pass or kernel level, and is therefore orthogonal to low-level kernel optimizations. Consistently, the relative improvement over the corresponding PPO baseline is nearly identical in plain PyTorch and verl (about +3.5% accuracy in both cases), confirming that the benefits of sparse-aware RL fine-tuning are complementary to, and composable with, existing framework-level optimizations. These results are summarized in Table 5.

Table 5: Framework extension results on Qwen2-7B (MMBench). AdaSparse-PRL is implemented on top of both plain PyTorch and verl, and compared against the corresponding PPO baselines.

Method	Framework	Accuracy (MMBench)	Peak Memory (GB)	Wall-clock Time (h)
PPO	Plain PyTorch	68.5	38.2	4.2
AdaSparse-PRL	Plain PyTorch	72.1	20.3	3.8
PPO	**verl + Flash Attn 2**	69.8	30.5	2.9
AdaSparse-PRL	**verl + Flash Attn 2**	**73.3**	**18.2**	**2.5**

G.2 Baseline Methods and Model Extensions

Accuracy Analysis. We further extend our evaluation to strong PEFT and RL baselines, including PPO, RFT, FARHF, EchoInk-R1, LoRA, QLoRA, GRPO, and Sparse FT, since AdaSparse-PRL is itself implemented on top of LoRA and is therefore complementary to, rather than competing with, parameter-efficient fine-tuning. We conduct experiments on the full Qwen3.5 series (2B, 4B, 9B, and 27B) across three datasets (MM-Vet, MMBench, and VQAv2), and report per-scale and average accuracy in Table 6. AdaSparse-PRL consistently outperforms all baselines at every model scale, with the largest gains observed on smaller models (e.g., +4.8% over PPO on Qwen3.5-2B), indicating particular benefits for resource-constrained scenarios. Averaged over the four scales,

AdaSparse-PRL reaches 72.1%, surpassing PPO (69.2%), RFT (68.5%), FARHF (69.8%), EchoInk-R1 (69.4%), LoRA (70.0%), QLoRA (69.6%), GRPO (70.7%), and Sparse FT (70.3%); notably, AdaSparse-PRL + QLoRA also attains a strong 71.7% average while using the smallest memory budget, confirming that our sparsity-aware design composes cleanly with PEFT. The effectiveness transfers from Qwen2 to Qwen3.5, confirming that our method is model-agnostic and not tied to any particular model family.

Table 6: Accuracy comparison across the Qwen3.5 series (2B, 4B, 9B, 27B) on multimodal benchmarks (MM-Vet, MMBench, VQAv2). Average denotes the mean over the four model scales.

Method	Qwen3.5-2B	Qwen3.5-4B	Qwen3.5-9B	Qwen3.5-27B	Average
PPO	62.3	67.2	71.5	75.6	69.2
RFT	61.6	66.6	70.8	74.9	68.5
FARHF	62.9	67.8	72.1	76.2	69.8
EchoInk-R1	62.6	67.4	71.7	75.8	69.4
LoRA	63.0	68.2	72.3	76.5	70.0
QLoRA	62.7	67.8	71.8	75.9	69.6
GRPO	63.9	68.9	73.0	77.1	70.7
Sparse FT	63.4	68.5	72.6	76.7	70.3
AdaSparse-PRL (Ours)	**65.1**	**70.4**	**74.5**	**78.5**	**72.1**
AdaSparse-PRL + QLoRA	64.8	70.0	74.0	78.1	71.7

Training Efficiency Analysis. On the efficiency side, we report training time (h), peak memory (GB), TFOPs (relative to PPO), and convergence steps for each Qwen3.5 scale in Tables 7–10. Across all four model sizes, AdaSparse-PRL reduces training time by 38–45%, peak memory by 21–35%, TFOPs by 48%, and convergence steps by 29% relative to PPO. For example, on Qwen3.5-9B, training time drops from 7.2 h (PPO) to 4.0 h, peak memory from 58.6 GB to 41.0 GB, TFOPs from 1.00× to 0.52×, and convergence steps from 1200 to 850. The gains are remarkably consistent across model scales, further validating the robustness of our approach; the AdaSparse-PRL + QLoRA variant pushes efficiency further (0.48× TFOPs, 820 steps), at a small accuracy cost, offering a favorable trade-off for extremely memory-constrained deployments.

Table 7: Training efficiency on Qwen3.5-2B. TFOPs are reported relative to the PPO baseline.

Method	Time (h)	Mem (GB)	TFOPs	Steps
PPO	2.1	18.5	1.00×	1200
RFT	2.0	18.2	0.98×	1200
FARHF	2.3	19.0	1.10×	1350
EchoInk-R1	2.5	19.5	1.20×	1400
LoRA	1.8	12.5	0.82×	1100
QLoRA	1.5	9.8	0.70×	1100
GRPO	2.0	18.5	0.95×	1000
Sparse FT	1.9	16.0	0.65×	1050
AdaSparse-PRL (Ours)	**1.3**	**12.0**	**0.52×**	**850**
AdaSparse-PRL + QLoRA	**1.2**	**9.5**	**0.48×**	**820**

Table 8: Training efficiency on Qwen3.5-4B. TFOPs are reported relative to the PPO baseline.

Method	Time (h)	Mem (GB)	TFOPs	Steps
PPO	3.8	32.4	1.00×	1200
RFT	3.7	32.0	0.98×	1200
FARHF	4.2	33.2	1.10×	1350
EchoInk-R1	4.5	34.0	1.20×	1400
LoRA	3.0	22.0	0.82×	1100
QLoRA	2.5	18.5	0.70×	1100
GRPO	3.6	32.4	0.95×	1000
Sparse FT	3.2	28.5	0.65×	1050
AdaSparse-PRL (Ours)	**2.1**	**21.5**	**0.52×**	**850**
AdaSparse-PRL + QLoRA	**2.0**	**17.0**	**0.48×**	**820**

Table 9: Training efficiency on Qwen3.5-9B. TFOPs are reported relative to the PPO baseline.

Method	Time (h)	Mem (GB)	TFOPs	Steps
PPO	7.2	58.6	1.00×	1200
RFT	7.0	58.0	0.98×	1200
FARHF	8.0	59.5	1.10×	1350
EchoInk-R1	8.5	60.5	1.20×	1400
LoRA	5.5	42.0	0.82×	1100
QLoRA	4.8	36.0	0.70×	1100
GRPO	6.8	58.6	0.95×	1000
Sparse FT	6.0	52.0	0.65×	1050
AdaSparse-PRL (Ours)	**4.0**	**41.0**	**0.52×**	**850**
AdaSparse-PRL + QLoRA	**3.8**	**34.0**	**0.48×**	**820**

Table 10: Training efficiency on Qwen3.5-27B. TFOPs are reported relative to the PPO baseline. † marks configurations requiring CPU offloading.

Method	Time (h)	Mem (GB)	TFOPs	Steps
PPO	18.5	78.4†	1.00×	1200
RFT	18.0	78.0†	0.98×	1200
FARHF	20.5	79.0†	1.10×	1350
EchoInk-R1	21.8	79.5†	1.20×	1400
LoRA	14.2	65.0	0.82×	1100
QLoRA	12.0	54.0	0.70×	1100
GRPO	17.5	78.4†	0.95×	1000
Sparse FT	15.5	72.0	0.65×	1050
AdaSparse-PRL (Ours)	**10.5**	**62.0**	**0.52×**	**850**
AdaSparse-PRL + QLoRA	**9.8**	**52.0**	**0.48×**	**820**