TinyAlign: Boosting Lightweight Vision-Language Models by Mitigating Modal Alignment Bottlenecks

Anonymous ACL submission

Abstract

Lightweight Vision-Language Models (VLMs) are indispensable for resource-constrained applications. The prevailing approach to aligning vision and language models involves freezing both the vision encoder and the language model while training small connector modules. However, this strategy heavily depends on the intrinsic capabilities of the language model, which can be suboptimal for lightweight models with limited representational capacity. In this work, we investigate this alignment bottleneck through the lens of mutual information, positing that the constrained capacity of the language model inherently limits the Effective Mutual Information (EMI) between multimodal inputs and outputs, thereby compromising alignment quality. To address this challenge, we propose TinyAlign, a novel framework inspired by Retrieval-Augmented Generation, which strategically retrieves relevant context from a memory bank constructed from training data to enrich multimodal inputs and enhance their alignment. Extensive empirical evaluations reveal that TinyAlign significantly reduces training loss, accelerates convergence, and enhances task performance with negligible computational overhead. Remarkably, it allows models to achieve baseline-level performance with only 40% of the fine-tuning data, highlighting exceptional data efficiency. Our work thus offers a practical pathway for developing more capable lightweight VLMs while introducing a fresh theoretical lens to better understand and address alignment bottlenecks in constrained multimodal systems.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have catalyzed the development of Vision-Language Models (VLMs), enabling models to excel in complex multimodal reasoning and understanding tasks. Prominent models such as Gemini 2.5 Pro (Google, 2024), GPT-4V (OpenAI, 2023), Qwen2.5-VL 72B (Bai et al., 2025), and PaLI-X (Chen et al., 2023b) have showcased remarkable performance across various benchmarks, setting new standards for multimodal intelligence. However, these models typically involve billions of parameters, resulting in significant computational and storage demands. Such massive requirements make them impractical for resource-constrained scenarios, such as edge devices or applications with limited computational budgets. This growing demand for efficiency has turned lightweight VLMs into a critical area of research, as they aim to retain strong multimodal capabilities while drastically reducing computational costs and memory footprints, thereby enabling broader applicability.

To achieve this balance between performance and efficiency, most lightweight VLMs adopt a modular design where pre-trained vision encoders and language models are frozen, and a small "connector" module is trained to align the two modalities. This approach, utilized by models such as MiniGPT-4 (Zhu et al., 2023), BLIP-2 (Li et al., 2023), and Visual Instruction Tuning (Liu et al., 2023), is computationally efficient and leverages the strong representational power of pre-trained components. However, while effective, this implicit alignment paradigm faces inherent challenges in the context of lightweight VLMs. The limited representational capacity of smaller LLMs significantly constrains their ability to process and align multimodal information, leading to subpar performance on complex tasks. This misalignment becomes a critical bottleneck for lightweight VLMs, preventing these models from fully realizing their potential.

To address this issue, we conduct a theoretical analysis of the alignment bottleneck in lightweight VLMs from the perspective of information theory(Zhu et al., 2024; Liu et al., 2025). Specifically, we introduce the concept of Effective Mutual Information (EMI) to quantify the amount of in-

formation a model can practically leverage given its capacity constraints. Our analysis posits that freezing the parameters of a lightweight language model inherently limits the EMI, creating a bottleneck that restricts the model's ability to align multimodal inputs effectively. This limitation manifests in suboptimal learning dynamics when using standard training objectives, highlighting the need for strategies to enhance the effective flow of information during alignment.

Motivated by this insight, we propose TinyAlign, a novel pre-training and fine-tuning framework explicitly designed to overcome this alignment bottleneck. Inspired by Retrieval-Augmented Generation (RAG) (Lewis et al., 2021; Guu et al., 2020; Hu et al., 2023), TinyAlign introduces a memory bank constructed directly from multimodal training instances within the dataset. This distinguishes our approach from traditional methods that depend on external knowledge sources. During training, the framework retrieves contextually relevant representations from the memory bank and augments the original visual inputs with enriched multimodal context. By increasing the effective information content available to the model, TinyAlign reduces the inherent learning difficulty posed by the limited capacity of lightweight language models. This design not only enhances alignment but also optimizes the utilization of available training data.

We validate the effectiveness of TinyAlign through extensive experiments across a diverse set of lightweight architectures, including Vicuna (Chiang et al., 2023), Phi-2 (Gunasekar et al., 2023), TinyLLaMA (Zhang et al., 2024), and Qwen2 (Yang et al., 2024), using vision encoders like SigLIP (Zhai et al., 2023) and CLIP (Radford et al., 2021). Our results demonstrate that TinyAlign significantly accelerates convergence, reduces alignment losses (Fig. 1(a)), and produces more compact and meaningful feature representations (Fig. 1(b)). Furthermore, TinyAlign exhibits exceptional data efficiency, achieving baseline-level performance with only 40% of the fine-tuning data. Crucially, this performance boost comes with negligible computational overhead ($\sim$0.3s latency increase and $\sim$2GB memory footprint), maintaining the efficiency required for lightweight deployment. Our contributions can be summarized as:

- We identify a fundamental alignment bottleneck in lightweight VLMs, theoretically analyzing its root cause using principles from information theory.
- We propose TinyAlign, a novel framework inspired by RAG, which enriches multimodal inputs by retrieving relevant context from a memory bank built from training data. This approach enhances the effective information content accessible to the model, addressing the alignment bottleneck.
- We conduct extensive experiments validating TinyAlign, demonstrating notable improvements in convergence speed, alignment quality, downstream task performance, and data efficiency, all while maintaining minimal computational overhead.

2 Related Work

The LLM-Centric Paradigm in VLMs. The LLM-Centric Paradigm has become a dominant framework in Vision-Language Models, leveraging pre-trained Large Language Models as the core for cross-modal understanding(Yang et al., 2024; Bai et al., 2025; Lu et al., 2024; Li et al., 2023; Chen et al., 2023a, 2024). This approach typically freezes the parameters of both the vision encoder and the LLM (Li et al., 2023), while training a lightweight connector module to bridge vision and language (Yang et al., 2024). This implicit alignment strategy has achieved notable empirical success, as seen in models like DeepSeek-VL (Lu et al., 2024) and Qwen2.5-VL (Bai et al., 2025). However, the theoretical mechanisms enabling effective cross-modal harmonization remain underexplored, with most research focusing on empirical results rather than systematic analysis. To fill this gap, our work conducts one of the first in-depth theoretical investigations into this paradigm, uncovering its principles and limitations to better understand cross-modal alignment.

Advancements in Lightweight VLMs. Recent efforts to create lightweight Vision-Language Models (VLMs) have explored various avenues for enhancing efficiency and performance(Yuan et al., 2024; Zhou et al., 2024; Yao et al., 2024; Marafioti et al., 2025; Steiner et al., 2024). EfficientVLM (Wang et al., 2022) introduces a distill-then-prune framework with modal-adaptive pruning to compress large VLMs effectively. TinyLLaVA (Zhou et al., 2024) explores optimal pairings of language models, vision encoders,

and connectors for small-scale VLMs, while MobileVLM and its successor v2 emphasize architectural innovations, high-quality data, and advanced training strategies (Chu et al., 2023, 2024), while SmolVLM (Marafioti et al., 2025) further explores tokenization strategies.MiniCPM-V (Yao et al., 2024) presents a series of efficient Multimodal Large Language Models (MLLMs) designed for on-device deployment, achieved by integrating advanced techniques in architecture, pre-training, and alignment. However, these approaches primarily focus on component optimization, model compression, advanced training strategies, or designing for edge deployment, and seldom question whether the widely adopted implicit alignment paradigm is fundamentally suitable for models with limited capacity. In contrast, we provide a principled analysis demonstrating that this paradigm intrinsically induces higher alignment loss for smaller models, thereby limiting their potential for robust visual understanding and cross-modal alignment.

Retrieval-Augmented Models. Retrieval-Augmented Generation enhances factual accuracy in NLP by integrating external knowledge retrieval with parametric models (Lewis et al., 2021). Techniques like unsupervised retriever pre-training enable efficient access to large-scale documents during training and inference (Guu et al., 2020). RAG now extends to multimodal tasks, with MM-REACT combining language models and vision experts for complex reasoning (Yang et al., 2023a), and Re-ViLM reducing parameters by storing knowledge externally for image-to-text generation (Yang et al., 2023b). Frameworks like RAVEN and MuRAG apply retrieval for multitask learning and open-domain question answering (Rao et al., 2024; Chen et al., 2022), while models like REVEAL unify memory, retrieval, and generation across diverse sources (Hu et al., 2023). These approaches rely on large external memory banks and retriever modules to broaden knowledge for reasoning-heavy tasks (Caffagni et al., 2024; Hu et al., 2023; Rao et al., 2024). In contrast, TinyAlign addresses the EMI bottleneck in lightweight VLMs by constructing a memory bank from multimodal *training instances*. During pre-training and fine-tuning, TinyAlign retrieves relevant representations to augment visual input, increasing mutual information between inputs and outputs and overcoming the limitations of compact models.

3 Theoretical Framework: An Information-Theoretic Perspective on Alignment

3.1 Cross-Entropy Loss in LLM-Centric VLM Pre-training

We begin by formalizing the standard LLM-centric paradigm for Vision-Language Models. In this setup, the objective is to align visual information with a pre-trained LLM. Let the visual input be denoted as X_V, the accompanying textual instruction as X_I, and the target output as L.

The processing pipeline in this standard paradigm is structured as follows:

1. A frozen Vision Encoder (e.g., ViT) with parameters θ_{ViT} extracts visual features: $Z_V = \text{ViT}(X_V; \theta_{\text{ViT}})$.
2. A trainable Connector module, parameterized by θ_C^*, transforms Z_V into embeddings H_V compatible with the LLM's input space: $H_V = \text{Connector}(Z_V; \theta_C^*)$.
3. The textual instruction X_I is processed by the frozen LLM's embedding layer (θ_{LLM}) to generate H_I.
4. The joint input is formed as $H_{\text{in}} = [H_V, H_I]$.
5. Finally, the frozen LLM produces an output distribution: $P_{\text{model}}(L|H_{\text{in}}; \theta_{\text{LLM}})$.

3.2 The Alignment Bottleneck: Irreducible Error and Effective Mutual Information

To understand the limitations of implicit alignment in lightweight models, we analyze the learning objective from an information-theoretic perspective. The standard training objective is minimizing the conditional cross-entropy (CE) loss, which can be decomposed into the true conditional entropy and the Kullback-Leibler (KL) divergence:

$$\begin{aligned}\mathcal{L}_{\text{CE}}(\theta_C^*) = H\big(P_{\text{true}}(L \mid X_V, X_I)\big) \\ + D_{\text{KL}}\Big(P_{\text{true}}(L \mid X_V, X_I) \,\Big\|\, \\ P_{\text{model}}(L \mid [H_V(\theta_C^*), H_I]; \theta_{\text{LLM}})\Big)\end{aligned} \tag{1}$$

Here, $H(P_{\text{true}})$ represents the inherent uncertainty in the data labels, which is independent of the model. Minimizing $\mathcal{L}_{\text{CE}}$ is thus equivalent to minimizing the KL divergence, which measures

the alignment between the model's predictions and the true distribution.

During training, the Connector θ_C^* learns to translate visual features Z_V into the LLM's semantic space. However, we **posit** that even with an optimal Connector θ_C^{opt}, the frozen LLM's fixed architecture and pre-trained knowledge impose a limit on how well it can interpret these foreign visual embeddings. We term this limitation the **Irreducible Alignment Error** under the standard paradigm:

$$\bar{\epsilon}_{\theta_{\text{LLM}}} = \mathbb{E}_{(X_V, X_I)} \left[\min_{\theta_C^*} \mathrm{D}_{\text{KL}}\big(P_{\text{true}} \,\|\, P_{\text{model}}(\dots; \theta_{\text{LLM}})\big) \right] \geq 0 \quad (2)$$

This term, $\bar{\epsilon}_{\theta_{\text{LLM}}}$, quantifies the "alignment gap"—the residual error that persists because the frozen LLM cannot perfectly assimilate the visual information provided solely through the connector. Consequently, the minimum achievable CE loss is bounded:

$$\min_{\theta_C^*} \mathcal{L}_{\text{CE}}(\theta_C^*) = H(L|X_V, X_I) + \bar{\epsilon}_{\theta_{\text{LLM}}} \quad (3)$$

To capture the system's practical capability, we introduce the concept of **Effective Mutual Information (EMI)**. EMI represents the mutual information the system can actually leverage after accounting for the capacity constraints of the frozen LLM:

$$I_{\text{eff}}(X_V, X_I; L|\theta_{\text{LLM}}) I(X_V, X_I; L) - \bar{\epsilon}_{\theta_{\text{LLM}}} \quad (4)$$

Substituting this into Eq. (3), the lower bound of the loss becomes:

$$\min_{\theta_C^*} \mathcal{L}_{\text{CE}}(\theta_C^*) \approx H(L) - I_{\text{eff}}(X_V, X_I; L|\theta_{\text{LLM}}) \quad (5)$$

This formulation suggests that the system's performance is fundamentally constrained by $\bar{\epsilon}_{\theta_{\text{LLM}}}$. A higher irreducible error directly reduces the EMI, hindering the model's ability to utilize multimodal inputs.

Hypothesis on LLM Scale. We hypothesize that $\bar{\epsilon}_{\theta_{\text{LLM}}}$ is dependent on the model scale. A smaller, lightweight LLM ($\theta_{\text{LLM,small}}$) likely possesses a more restricted representational capacity compared to a larger LLM ($\theta_{\text{LLM,large}}$). As empirically observed in Fig. 2(a), larger models exhibit lower initial loss (likely due to stronger language priors), suggesting a greater capacity to handle semantic information. This implies:

$$\bar{\epsilon}_{\theta_{\text{LLM,small}}} > \bar{\epsilon}_{\theta_{\text{LLM,large}}}$$

Consequently, lightweight VLMs suffer from lower Effective Mutual Information (I_{eff}) and a higher minimum loss bound. This theoretical perspective motivates our proposed method: since we cannot easily reduce $\bar{\epsilon}_{\theta_{\text{LLM}}}$ within the standard paradigm (due to the frozen, lightweight LLM), we must alter the input paradigm to facilitate easier alignment.

4 TinyAlign: Mitigating Modal Alignment Bottlenecks in Lightweight VLMs

4.1 Theoretical Analysis: Enhancing Effective Mutual Information via RAG

As analyzed in Sec. 3, lightweight frozen LLMs often suffer from a high irreducible alignment error $\bar{\epsilon}_{\theta_{\text{LLM}}}$ under the standard paradigm. TinyAlign (Fig. 1) mitigates this by fundamentally altering the input paradigm. We introduce a Retrieval-Augmented Generation (RAG) mechanism to boost the *Effective Mutual Information* (I_{eff}) by supplying strategically compressed, relevant contextual cues.

A standard VLM maps visual input X_V to H_V via a primary connector. TinyAlign augments this process by: 1) retrieving k pre-compressed embeddings E_R from a memory bank $\mathcal{M}$; 2) transforming E_R into supplementary representations H_R via a trainable RAG connector θ_{RC}^*; and 3) presenting a composite input $H'_{\text{in}} = [H_V, H_R, H_I]$ to the frozen LLM. We **hypothesize** that incorporating E_R (forming augmented context $X' = (X_V, E_R, X_I)$) increases the effective mutual information I_{eff}. The improvement, ΔI_{eff}, can be decomposed as:

$$\begin{aligned} \Delta I_{\text{eff}} &= [I(X'; L) - \bar{\epsilon}_{\theta_{\text{LLM}}}(X')] \\ &\quad - [I(X_V, X_I; L) - \bar{\epsilon}_{\theta_{\text{LLM}}}(X_V, X_I)] \\ &= \underbrace{I(E_R; L|X_V, X_I)}_{\Delta I_{\text{true}}} \\ &\quad + \underbrace{(\bar{\epsilon}_{\theta_{\text{LLM}}}(X_V, X_I) - \bar{\epsilon}_{\theta_{\text{LLM}}}(X'))}_{\Delta \bar{\epsilon}_{\text{reduction}}} \end{aligned} \quad (6)$$

Here, $\Delta I_{\text{true}} > 0$ represents the novel information provided by the retrieved examples (e.g.,

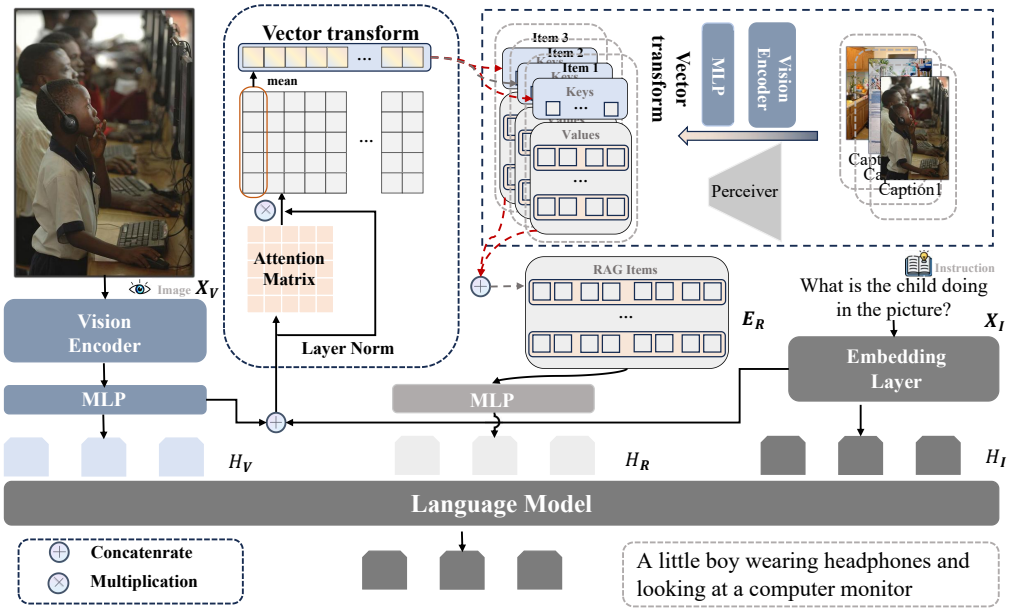


Figure 1: Architectural overview of TinyAlign. Given an input image X_V and instruction X_I, a query key derived from these inputs retrieves k similar, Perceiver-compressed multimodal embeddings $E_R = \{E_{R_j}\}_{j=1}^{k}$ from a pre-constructed Memory Bank (built from training data). These cues E_R are processed by a trainable RAG Connector (θ^*_{RC}) into an auxiliary representation H_R. Concurrently, X_V is processed by a Vision Transformer (ViT) and a primary Connector (θ^*_C) into visual features H_V. The instruction X_I is embedded as H_I. Finally, a frozen LLM receives the composite input $H'_{\text{in}} = [H_V, H_R, H_I]$. This architecture enhances lightweight VLMs by supplying efficiently processed, relevant contextual information, thereby alleviating the alignment burden.

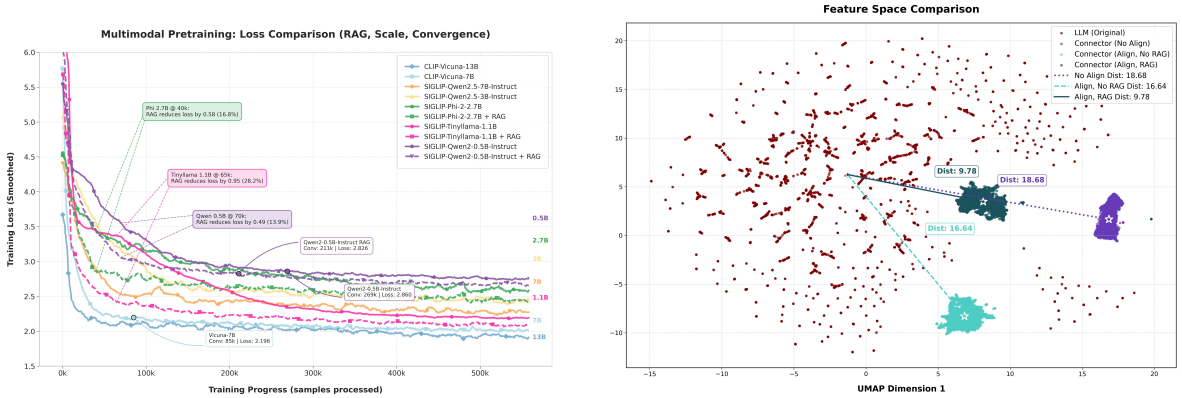
relevant captions). More importantly, the second term, $\Delta\bar{\epsilon}_{\text{reduction}}$, represents the reduction in alignment difficulty. By transforming E_R into "LLM-assimilable contextual hints" via the RAG connector, we provide the LLM with information in a format it can more easily process than raw visual embeddings. This makes the input H'_{in} more aligned with the LLM's pre-trained priors, effectively lowering the irreducible error for the augmented task ($\bar{\epsilon}_{\theta_{\text{LLM}}}(X') < \bar{\epsilon}_{\theta_{\text{LLM}}}(X_V, X_I)$). This reduction is particularly crucial for lightweight VLMs with limited intrinsic reasoning capacity. TinyAlign acts as a cognitive scaffold, lowering the threshold for effective alignment. Consequently, the minimum achievable CE loss is reduced, as corroborated by the accelerated convergence observed in our experiments (Fig. 2(a)).

4.2 Memory Bank Design for Efficiency

To ensure practical deployability on resource-constrained devices, TinyAlign's memory bank $\mathcal{M}$ is designed for minimal storage and low latency. Crucially, the memory bank is constructed entirely from internal multimodal training instances, avoiding reliance on external knowledge bases. We sample 100K image-text pairs from the pre-training dataset (Sec. 5.1) to populate the bank.

Key Generation. Each key $K_m \in \mathbb{R}^{d_k}$ is a compact embedding derived from a source image-text pair (X_{V_m}, X_{I_m}) via an attention-based aggregation mechanism. This distills salient cross-modal information into a dense vector, facilitating rapid similarity search using maximum inner product search (MIPS).

Value Generation (Perceiver Compression). For the values V_m, we employ an LLM-independent Perceiver (Jaegle et al., 2022) model (θ_P) to pre-process original multimodal instances into compressed latent embeddings E_{R_m}. **Design Rationale:** While retrieving raw text is possible, it incurs significant computational overhead during inference due to the variable and potentially long sequence lengths. The Perceiver model compresses high-dimensional multimodal inputs into a small, fixed number of latent tokens (e.g., 32 latents). This fixed-size representation ensures that augmenting the input with k retrieved examples incurs negligi-



(a) Loss Dynamics (b) UMAP Space

Figure 2: **(a)** Comparison of multimodal pre-training loss on the LLaVA dataset. Larger models exhibit lower initial loss due to stronger text priors, yet TinyAlign-enhanced models (dashed lines) consistently accelerate convergence and achieve lower final loss. **(b)** UMAP visualization showing TinyAlign promotes superior semantic clustering compared to baseline, reducing the modality gap.

ble latency and memory overhead (see Appendix D for a detailed comparison with text-only retrieval), making it highly suitable for lightweight VLMs.

4.3 Integrated Pre-training and Instruction Tuning

TinyAlign employs a two-stage strategy consistent with standard VLM protocols, both optimizing the objective in Eq. (5):

Stage 1: Connector Pre-training. The vision encoder θ_{ViT} and LLM $\theta_{\text{LLM,small}}$ are frozen. Only the connectors θ_C^* and θ_{RC}^* are trained. This forces the RAG connector to learn how to optimally format the retrieved E_R into the LLM's semantic space, validating the "scaffolding" hypothesis.

Stage 2: Instruction Tuning. The ViT remains frozen, while the connectors and the lightweight LLM are fine-tuned. Here, the LLM adapts to maximally leverage the augmented context H_R provided by the active memory bank $\mathcal{M}$ for downstream tasks.

5 Experiments

5.1 Experimental Setup

Our framework builds upon lightweight VLMs. We use the LLaVA pre-training set (558K pairs) for pre-training and memory bank construction, where we sample 100k pairs to build the bank. For instruction tuning, we use LLaVA v1.5 SFT (665K samples). We validate generalization across multiple model scales, including Qwen2-0.5B, TinyLLaMA-1.1B, Phi-2 (2.7B), and Qwen2.5-3B. Detailed hyperparameters are provided in Appendix B.

5.2 Pre-training Performance Analysis

TinyAlign accelerates convergence noticeably. As shown in Fig. 2(a), Phi-2 (2.7B) achieves a 16.8% loss reduction at comparable training steps, indicating faster and more stable optimization under the same budget.

We also observe a pronounced text-only bias during early training. Larger models (e.g., Phi-2 compared to Qwen2-0.5B) often start with lower initial loss even before receiving any visual input. This suggests that stronger intrinsic language priors allow the model to produce plausible tokens without visual grounding, which can create an early "plateau" driven by language modeling alone. TinyAlign mitigates this effect by injecting explicit, relevant multimodal context via retrieval, enabling the model to move beyond the language-driven plateau and align visual modalities substantially faster than standard connectors.

The learned embedding space is also improved. UMAP projections in Fig. 2(b) show tighter and more semantically coherent clusters with TinyAlign, consistent with reduced alignment error $\bar{\epsilon}_{\theta_{\text{LLM}}}$.

5.3 Instruction Tuning Performance Analysis

Table 1 summarizes performance across multimodal benchmarks.

TinyAlign yields consistent gains from 0.5B to 3B parameters. Although absolute improvements can vary by model size (e.g., the gain on VQAv2 is smaller for Qwen2.5-3B than for Phi-2), the overall trend remains stable. This supports the view that even when larger models have lower irreducible error, an alignment bottleneck still persists, and

Table 1: Performance comparison on multimodal benchmarks. ↑indicates improvement. Note that all results are verified with statistical significance tests (95% Confidence Interval), confirming robust improvements.

Benchmark	Qwen2-0.5B		TinyLLaMA-1.1B		Phi-2-2.7B		Qwen2.5-3B	
	Base	+TA	Base	+TA	Base	+TA	Base	+TA
GQA	56.3	**57.6** ↑	52.4	**56.7** ↑	58.4	**60.7** ↑	60.2	**62.6** ↑
MMMU	31.0	**31.4** ↑	29.4	**30.2** ↑	36.2	**37.7** ↑	37.2	**38.8** ↑
MM-Vet	20.9	**23.6** ↑	25.1	**26.8** ↑	31.8	**33.4** ↑	32.4	**34.4** ↑
POPE	86.4	**87.2** ↑	84.3	**86.0** ↑	86.6	**88.0** ↑	85.1	**87.8** ↑
SQA-I	59.5	**60.2** ↑	56.5	**59.8** ↑	67.3	**68.1** ↑	70.2	**71.6** ↑
TextVQA	46.1	**46.6** ↑	46.3	**46.4** ↑	50.3	**55.5** ↑	54.8	**57.1** ↑
VQAV2	73.0	**74.2** ↑	71.0	**74.5** ↑	75.4	**78.3** ↑	79.6	**81.2** ↑
MME	1171	**1209** ↑	1105	**1201** ↑	1364	**1412** ↑	1401	**1442** ↑

Table 2: Mechanism Validation on Phi-2. TinyAlign offers gains distinct from parameters, data scaling, or static augmentation.

Configuration	Description	Data	Score
1. Baseline	MLP	Original	70.5
2. Wider-Conn	∼6M Params	Original	71.2
3. Static RAG	Dataset Augmentation	Original	72.8
4. TinyAlign	**Dynamic RAG-Conn**	**Original**	**73.3**
5. Base+Data	MLP	+ShareGPT	73.7
6. TA+Data	**Dynamic RAG-Conn**	**+ShareGPT**	**74.1**

TinyAlign effectively alleviates it up to the 3B scale.

Performance improvements extend to complex reasoning benchmarks as well. Statistical significance testing with a 95% confidence interval on MM-Vet confirms robust gains (e.g., TinyLLaMA improves from 25.1 to 26.8), suggesting that retrieval-augmented alignment does not degrade reasoning ability and can improve it under controlled evaluation.

We additionally compare TinyAlign (Phi-2) against a standard end-to-end finetuning baseline. TinyAlign achieves a higher average score (73.3) than end-to-end finetuning (70.2). This difference highlights a practical advantage for lightweight models: end-to-end finetuning can more easily overfit or induce catastrophic forgetting due to limited parameter capacity, whereas TinyAlign provides external contextual scaffolding that guides a frozen LLM, preserving general language capability while enhancing multimodal alignment.

5.4 Mechanism Validation

To isolate the source of the improvements, we conduct controlled experiments summarized in Table 2.

First, the gains cannot be explained by adding parameters alone. A wider-connector baseline that matches TinyAlign's additional capacity (approximately 6M parameters) yields only a minor improvement (71.2 vs. 70.5), while TinyAlign reaches 73.3, indicating that the retrieval mechanism is the key contributor rather than projection-layer capacity.

Second, the gains are not simply a result of more data. When pre-training is augmented with additional high-quality ShareGPT4V data, the baseline reaches 73.7. TinyAlign remains additive: combining TinyAlign with the extra data achieves 74.1. This suggests that TinyAlign improves the learning mechanism by easing alignment, and its benefits are complementary to data scaling.

Third, dynamic retrieval is more effective than static augmentation. A static RAG baseline that augments training samples with retrieved captions improves over the baseline (72.8 vs. 70.5), but TinyAlign performs better (73.3). This indicates that query-specific, dynamic retrieval through a dedicated pathway is more beneficial than static augmentation, particularly because TinyAlign continues to assist the frozen LLM during inference by providing real-time contextual guidance.

Finally, Perceiver compression is critical for practical deployment. Replacing Perceiver-compressed multimodal cues with naive text-only retrieval (using LLM token embeddings directly) can yield modest gains due to strong text understanding, but it incurs prohibitive overhead: inference latency increases by roughly 21× (0.2s → 4.2s) and memory usage by about 20× (2GB → 40GB) due to long concatenated text sequences. In contrast, the Perceiver design achieves comparable improvements with negligible overhead (approximately 0.3s latency), making retrieval-augmented alignment feasible for lightweight systems.

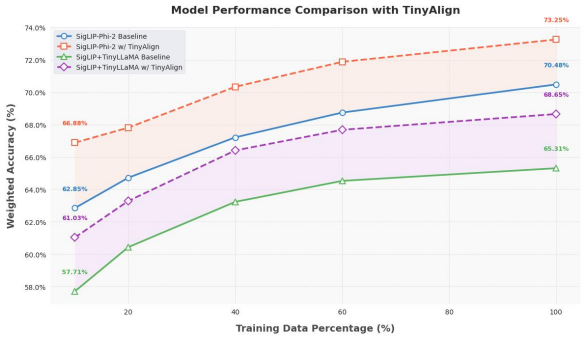


Figure 3: Model weighted accuracy vs. instruction tuning data percentage. TinyAlign matches full-data baseline performance with only 40% data.

5.5 Data Efficiency Analysis

TinyAlign demonstrates strong data efficiency, as shown in Fig. 3. Models equipped with TinyAlign match the full-data baseline performance using only 40% of the instruction tuning data, indicating that retrieval-augmented context can compensate for reduced supervised signal in fine-tuning.

We further examine robustness to domain shift by using an out-of-domain (OOD) memory bank built from ShareGPT4V. Even with OOD retrieval, TinyAlign achieves 71.0 compared to the baseline 70.5, while in-domain retrieval remains best (73.3). This suggests the architecture provides a generic benefit by making alignment easier (acting as a structural prior), and domain-relevant cues provide additional gains by offering more precise semantic bridges.

5.6 Ablation Study

We conduct targeted ablations to finalize the design. A 100k-entry memory bank is sufficient (Table 6). Matching the vision encoder used for VLM training and for RAG key generation is important for stable retrieval and performance (Table 7). We also ablate retrieval count and find that Top-5 provides the best balance between context enrichment and noise, as verified on Phi-2 and TinyLLaMA.

6 Conclusion

Lightweight Vision-Language Models are important for resource-constrained applications, but their performance is often limited by alignment bottlenecks caused by the restricted capacity of smaller language models. From a mutual information perspective, this limitation reduces the Effective Mutual Information between multimodal inputs and outputs, which in turn degrades alignment quality. To address this issue, we propose TinyAlign, a framework inspired by Retrieval-Augmented Generation that retrieves relevant context from a memory bank to enrich multimodal inputs. Empirical results show that TinyAlign improves task performance, reduces training loss, and accelerates convergence. Notably, it reaches baseline-level performance using only 40% of the fine-tuning data, offering a highly data-efficient solution for lightweight VLMs. Overall, this work provides a practical framework and a theoretical lens for addressing alignment challenges in constrained multimodal systems.

7 Limitations

Though TinyAlign is effective, it relies on a well-designed memory bank and compatibility between the vision encoder and retrieval keys. Additionally, while we validated generalization up to 3B parameters, the gains may diminish for extremely large LLMs (>30B) where alignment bottlenecks are less severe.

8 Future work

(1) **Extension to other modalities:** Applying the TinyAlign framework to Audio-Text or Video-Text alignment, where lightweight models similarly struggle with high-dimensional, temporal inputs. Our framework's ability to compress context via Perceiver makes it uniquely suited for these data-intensive modalities.

(2) **Adaptive Retrieval:** Developing mechanisms to dynamically decide when to retrieve (e.g., based on model confidence), further optimizing inference efficiency by only invoking the memory bank for ambiguous or complex queries.

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, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Davide Caffagni, Federico Cocchi, Nicholas Moratelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2024. Wiki-llava: Hierarchical retrieval-augmented generation for multimodal llms. *Preprint*, arXiv:2404.15406.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *Preprint*, arXiv:2310.09478.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. *Preprint*, arXiv:2210.02928.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, and 24 others. 2023b. Pali-x: On scaling up a multilingual vision and language model. *Preprint*, arXiv:2305.18565.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *Preprint*, arXiv:2312.14238.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, and Chunhua Shen. 2023. Mobilevlm : A fast, strong and open vision language assistant for mobile devices. *Preprint*, arXiv:2312.16886.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and Chunhua Shen. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *Preprint*, arXiv:2402.03766.

Google. 2024. Gemini 2.5 pro. `https://deepmind.google/technologies/gemini/`. Google DeepMind.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *Preprint*, arXiv:2002.08909.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. 2023. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. *Preprint*, arXiv:2212.05221.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. 2022. Perceiver io: A general architecture for structured inputs & outputs. *Preprint*, arXiv:2107.14795.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Tianyu Liu, Jirui Qi, Paul He, Arianna Bisazza, Mrinmaya Sachan, and Ryan Cotterell. 2025. Pointwise mutual information as a performance gauge for retrieval-augmented generation. *Preprint*, arXiv:2411.07773.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. 2025. Smolvlm: Redefining small and efficient multimodal models. *Preprint*, arXiv:2504.05299.

OpenAI. 2023. Gpt-4v(ision). `https://openai.com/research/gpt-4v-system-card`. OpenAI.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Varun Nagaraj Rao, Siddharth Choudhary, Aditya Deshpande, Ravi Kumar Satzoda, and Srikar Appalaraju. 2024. Raven: Multitask retrieval augmented vision-language learning. *Preprint*, arXiv:2406.19150.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. 2024. Paligemma 2: A family of versatile vlms for transfer. *Preprint*, arXiv:2412.03555.

Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. 2022. Efficientvlm: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. *arXiv preprint arXiv:2210.07795*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023a. Mm-react: Prompting chatgpt for multimodal reasoning and action. *Preprint*, arXiv:2303.11381.

Zhuolin Yang, Wei Ping, Zihan Liu, Vijay Korthikanti, Weili Nie, De-An Huang, Linxi Fan, Zhiding Yu, Shiyi Lan, Bo Li, Ming-Yu Liu, Yuke Zhu, Mohammad Shoeybi, Bryan Catanzaro, Chaowei Xiao, and Anima Anandkumar. 2023b. Re-vilm: Retrieval-augmented visual language model for zero and few-shot image captioning. *Preprint*, arXiv:2302.04858.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, and 4 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *Preprint*, arXiv:2408.01800.

Zhengqing Yuan, Zhaoxu Li, Weiran Huang, Yanfang Ye, and Lichao Sun. 2024. Tinygpt-v: Efficient multimodal large language model via small backbones. *Preprint*, arXiv:2312.16862.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. 2024. Tinyllava: A framework of small-scale large multimodal models. *Preprint*, arXiv:2402.14289.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *Preprint*, arXiv:2304.10592.

Kun Zhu, Xiaocheng Feng, Xiyuan Du, Yuxuan Gu, Weijiang Yu, Haotian Wang, Qianglong Chen, Zheng Chu, Jingchang Chen, and Bing Qin. 2024. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. *Preprint*, arXiv:2406.01549.

A Detailed Analysis on Enhancing Effective Mutual Information via RAG

As discussed in the main text, lightweight, frozen LLMs ($\theta_{\text{LLM,small}}$) often exhibit performance limitations due to a substantial irreducible alignment error $\bar{\epsilon}_{\theta_{\text{LLM}}}$. To mitigate this, we introduce TinyAlign, a RAG-enhanced connector architecture. This approach boosts *effective mutual information* (I_{eff}) by supplying strategically compressed, highly relevant contextual cues.

A standard VLM processes a visual input X_V via a ViT (θ_{ViT}) to obtain Z_V, which a primary connector (θ_C^*) maps to H_V. Instruction embeddings H_I are also generated. TinyAlign augments this by: 1) retrieving k relevant, pre-compressed embeddings $E_R = \{E_{R_j}\}_{j=1}^k$ from a memory bank $\mathcal{M}$; 2) employing a trainable RAG connector (θ_{RC}^*) to transform E_R into supplementary representations H_R; and 3) presenting a composite input $H'_{\text{in}} = [H_V, H_R, H_I]$ to the frozen LLM.

We **posit** that incorporating E_R—forming an augmented context $X' = (X_V, X_I, E_R)$—enhances $I_{\text{eff}}(X'; L|\theta_{\text{LLM}}, \theta_{\text{ViT}})$. The change, ΔI_{eff}, is decomposed as:

$$\begin{aligned}\Delta I_{\text{eff}} &= [I(X'; L) - \bar{\epsilon}_{\theta_{\text{LLM}}}(X')] \\ &\quad - [I(X_V, X_I; L) - \bar{\epsilon}_{\theta_{\text{LLM}}}(X_V, X_I)] \\ &= \underbrace{I(E_R; L|X_V, X_I)}_{\Delta I_{\text{true}}} \\ &\quad + \underbrace{(\bar{\epsilon}_{\theta_{\text{LLM}}}(X_V, X_I) - \bar{\epsilon}_{\theta_{\text{LLM}}}(X'))}_{\Delta \bar{\epsilon}_{\text{reduction}}} \end{aligned} \quad (7)$$

The first term, ΔI_{true}, is positive because E_R, derived from pertinent captions, provides novel information about L. The second term, $\Delta \bar{\epsilon}_{\text{reduction}}$, signifies a positive reduction in the alignment difficulty. The RAG connector θ_{RC}^* transforms E_R into 'LLM-assimilable contextual hints.' These hints present information in a format more attuned to the LLM's textual processing strengths than deciphering complex visual semantics solely from H_V. This enhanced 'input friendliness' enables the fixed-capacity LLM to approximate the target distribution with greater fidelity, effectively lowering the irreducible error.

B Hyperparameter Summary

This subsection provides a comprehensive overview of the critical hyperparameters employed throughout our experimental phases. Table 3 delineates the comparative settings for pre-training and fine-tuning. Complementing this, Table 4 itemizes the architectural hyperparameters of the Perceiver model.

Table 3: Key hyperparameters for pre-training and fine-tuning.

Hyperparameter	Pre-training	Fine-tuning
Global Batch Size	256	128
Per-device Batch Size	16	12
Gradient Accumulation	1	2
Learning Rate	1e-3	5e-8
LR Scheduler	Cosine	Cosine
Warmup Ratio	0.03	0.03
Precision	FP16	FP16
Optimizer	AdamW	AdamW
LLM Tuning	Frozen	Full
Vision Tower Tuning	Frozen	Frozen
Connector Tuning	Full	Full

Table 4: PerceiverConfig Hyperparameters

Parameter	**Value**
`num_latents`	32
`d_latents`	96
`d_model`	128
`num_self_attends_per_block`	8
`num_blocks`	1
`num_self_attention_heads`	8
`num_cross_attention_heads`	8
`qk_channels`	96
`v_channels`	96
`image_size`	384

C UMAP Visualization Details

To elucidate the latent structure, we employed Uniform Manifold Approximation and Projection (UMAP). **Methodology:** The process utilized two sets of high-dimensional feature vectors: (1) **Connector Features**: Vectors derived from images post-processing by the vision tower and connector. (2) **LLM Input Embeddings**: Vectors representing embeddings of textual inputs. These sets were concatenated, reduced to 2D via UMAP, and visualized. As shown in the main text (Fig. 2b), TinyAlign brings visual features significantly closer to the text embedding space, reducing the modality gap.

D Efficiency Analysis: Perceiver vs. Text-Only

This section provides a quantitative breakdown of the computational efficiency of TinyAlign compared to a naive Text-Only retrieval baseline (Table 5).

Table 5: Efficiency Comparison. While Text-Only Retrieval yields slightly higher accuracy, it incurs prohibitive latency ($21\times$) and memory ($20\times$) costs due to long sequence lengths. TinyAlign (Perceiver) offers the best trade-off.

Method	Latency (s)	Memory (GB)	Avg. Score
Baseline	0.18	-	70.5
Text-Only RAG	4.20 ($21\times$)	40 ($20\times$)	74.7
TinyAlign	**0.21** ($1.1\times$)	**2**	**73.3**

Analysis: The Text-Only approach feeds raw retrieved captions directly into the LLM. While this leverages the LLM's text processing power (score +1.4%), the input sequence length explodes, causing massive latency and memory spikes. TinyAlign uses Perceiver IO to compress this information into fixed-size tokens (32 latents), maintaining near-baseline efficiency with comparable performance gains.

E Detailed Data Efficiency Analysis

TinyAlign models exhibit remarkable data efficiency. As shown in Fig. 4, TinyAlign-enhanced models consistently outperform baselines across varying data fractions.

F Additional Ablation Studies

Knowledge Base Size. We evaluate KB sizes of 100k, 300k, and 500k entries (Table 6). The 100k KB shows comparable performance to larger KBs while offering greater efficiency, justifying our choice.

Table 6: Ablation on KB size (Phi-2).

KB Size	GQA	MM-Vet	VQAv2	Avg
100k	60.7	34.1	78.3	**73.3**
300k	61.3	31.7	78.6	73.1
500k	61.4	31.6	78.6	72.9

Vision Encoder Alignment. We confirm that the vision encoder used for VLM training must match the one used for generating RAG keys (Table 7). Mismatched encoders lead to performance collapse.

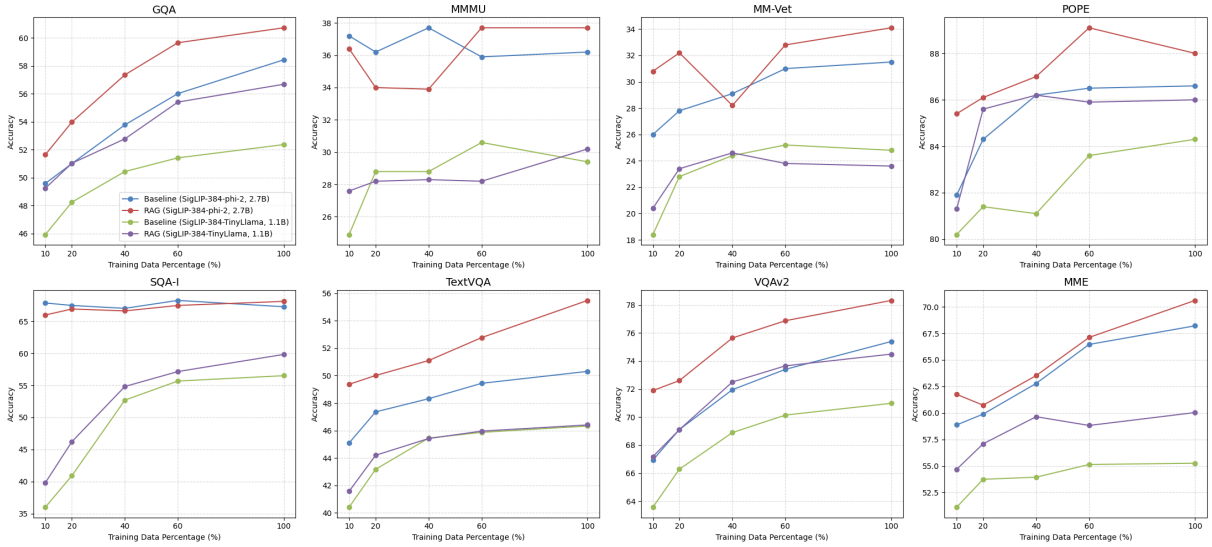


Figure 4: Detailed data efficiency analysis across individual benchmarks. Performance of TinyAlign-enhanced models is compared against baselines at varying percentages of instruction tuning data.

Table 7: Ablation on vision encoder alignment.

Benchmark	Matched (SigLIP)	Mismatched (CLIP)
GQA	51.28	40.96
MM-Vet	28.4	24.6
VQAv2	70.12	29.62
MME	1218.4	1054.8

Top-K Retrieval on TinyLLaMA. While Phi-2 showed lower sensitivity to Top-K, we conducted an additional ablation on TinyLLaMA (Table 8). The results confirm that **Top-5 retrieval** provides the optimal balance, avoiding the noise introduced by Top-10 while providing more context than Top-1.

Table 8: Ablation on Top-K retrieval for TinyLLaMA-1.1B. Top-5 yields the best performance.

Top-K	Avg. Score
Top-1	63.1
Top-5	**68.7**
Top-10	65.3

Table 9: Impact of Memory Bank Source (Phi-2).

Memory Source	Domain	Avg. Score
None (Baseline)	-	70.5
ShareGPT4V	Out-of-Domain	71.0
LLaVA (Ours)	**In-Domain**	**73.3**

Robustness: Out-of-Domain Memory Bank. To test if TinyAlign relies solely on in-domain data, we constructed a memory bank using Out-of-Domain (OOD) samples from ShareGPT4V (Table 9). Even with OOD retrieval, TinyAlign (71.0) outperforms the baseline (70.5), demonstrating the architectural benefit of the RAG pathway. However, In-Domain retrieval (73.3) remains optimal.