Fill the GAP: A Granular Alignment Paradigm for Visual Reasoning in Multimodal Large Language Models

Yanting Miao[*1,2,4], **Yutao Sun**[*1,3], **Dexin Wang**[1], **Mengyu Zhou**[†1], **Pascal Poupart**[2,4], **Lei Lv**[1], **Qi Zhao**[1], **Li Wang**[1], **Hao Li**[1], **Xiaoxi Jiang**[1] **and Guanjun Jiang**[1]

[1]Qwen Business Unit, Alibaba, [2]University of Waterloo,
[3]Zhejiang University, [4]Vector Institute

[*]Work done during an internship at Alibaba.
[†]Corresponding author: Mengyu Zhou (zhoumengyu.zmy@alibaba-inc.com).

Abstract

Visual latent reasoning lets a multimodal large language model (MLLM) create intermediate visual evidence as continuous tokens, avoiding external tools or image generators. However, existing methods usually follow an output-as-input latent paradigm and yield unstable gains. We identify a feature-space mismatch behind this instability: dominant visual-latent models build on pre-norm MLLMs and reuse decoder hidden states as predicted latent inputs, even though these states occupy a substantially different norm regime from the input embeddings the model was trained to consume [25, 12, 20]. This mismatch makes direct latent feedback unreliable. Motivated by this diagnosis, we propose **GAP**, a **G**ranular **A**lignment **P**aradigm for visual latent modeling. GAP aligns visual latent reasoning at three levels: data-level alignment grounds latent targets with inspectable auxiliary visual supervision; feature-level alignment maps decoder outputs into input-compatible visual latents through a lightweight PCA-aligned latent head; and model-level alignment assigns latent supervision selectively to examples where the base MLLM struggles. On Qwen2.5-VL 7B, GAP achieves the best aggregate perception and reasoning scores among all compared methods and is the only evaluated latent method that improves both over the base model. Inference-time interventions further show that generated latents carry task-relevant visual signal rather than merely adding token slots. Code is available at https://github.com/Qwen-Applications/GAP.

1 Introduction

Multimodal large language models (MLLMs) increasingly solve visual tasks by interleaving perception and language reasoning, yet many failures on charts, high-resolution perception, and visual math stem from missing or poorly localized visual evidence [1, 17, 19]. A natural way to obtain such evidence is to call external visual tools, such as cropping, search, or image generation [14], to create auxiliary views. Tool-augmented pipelines, however, introduce extra API calls, wall-clock latency, and system complexity, and building a unified generation-and-understanding model is a substantially heavier direction that typically requires many additional visual tokens and training resources. Tool-based approaches also face cross-task generalization difficulties, since a single visual tool is rarely adequate across diverse visual reasoning tasks [18, 30]. We instead ask whether an image-understanding MLLM can be minimally adapted to produce intermediate visual evidence inside its own autoregressive computation. One appealing solution is *visual latent reasoning*: the model emits continuous latent tokens intended to represent auxiliary visual information [21, 9]. This

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

design is lightweight and keeps inference inside a single model, with no external image-generation call. However, prior latent-token methods deliver unstable empirical gains, and in some settings the latent channel contributes little after post-training [10, 24, 13].

A feature-level failure mode in pre-norm MLLMs. Modern MLLMs such as Qwen3-VL [1], Gemma 4 [5], and InternVL3.5 [22] use pre-norm transformers, in which final-layer hidden states accumulate much larger norms than the input text or vision embeddings and are not guaranteed to lie in the same empirical subspace as input vision embeddings. Most recent visual-latent models build on pre-norm MLLM backbones and reuse output hidden states as next-step latent inputs [21, 9], implicitly assuming that these outputs are valid inputs. We test this assumption on Monet-7B [21], a state-of-the-art latent model based on Qwen2.5-VL-7B and find that the state it feeds back as the next latent is still about $4.7\times$ larger in norm than the input vision embeddings, even after the backbone's final RMSNorm (Figure 2(b)). Re-injecting such states creates a feature-space mismatched feedback loop, in which every latent step drives the model with inputs from a distribution it was not trained to consume.

Interventional evidence. The norm gap is not only descriptive. Taking Monet-7B as an output-as-input latent baseline and applying a training-free Exponential Moving Average (EMA) rescaling that matches predicted latent norms to vision input-embedding norms improves performance by $+2.00$ on MathVista. The intervention changes neither the training data nor the backbone; it only corrects the norm of the re-injected latent vector. This supports feature-level compatibility as a practical factor for stable latent reuse. Even without changing data or backbone, correcting the re-injected latent's magnitude already recovers measurable accuracy, and the residual gap motivates the learned alignment we introduce next.

Our approach: GAP. The feature-level mismatch above is the main failure mode targeted in this work, and norm rescaling alone treats only one symptom of it. Training a useful latent channel also benefits from two practical supports: the latent targets should be interpretable enough to inspect during training and analysis, and latent supervision should not be applied indiscriminately to examples the base model already solves. We therefore introduce **GAP**, a **G**ranular **A**lignment **P**aradigm for visual latent modeling across data, feature, and model levels. At the feature level, a **PCA-aligned latent head** maps the backbone's native-RMS-normalized decoder states into the principal subspace of auxiliary-image vision embeddings before re-injection; the PCA basis acts both as a subspace projector that returns generated latents to the empirical vision-embedding distribution and as a low-rank parameterization of the latent head. For inspectability, **context-grounded latent supervision** annotates each latent example with chain-of-thought intent, a `<latent>` span supervised by auxiliary-image vision embeddings, and a `<parser>` description that records the intended auxiliary visual signal during training and analysis. For selective supervision, **difficulty-aware latent assignment** estimates base-model accuracy by repeated sampling and assigns latent targets only to examples the base model fails to solve, while retaining text supervision on easier examples. Figure 1 summarizes GAP as a three-level alignment recipe spanning data-, feature-, and model-level alignment. The resulting training set contains 49K curated multimodal examples spanning visual CoT, charts, geometry, multimodal math, counting, and visual search.

We make the following contributions:

1. **A granular alignment paradigm (GAP).** We introduce a lightweight latent feedback recipe centered on a PCA-aligned projection that returns generated latents to the vision-embedding subspace, supported by context-grounded latent supervision for inspectability and difficulty-aware latent assignment for selective supervision.

2. **Empirical validation of aligned visual latents.** Across five perception and reasoning benchmarks with a Qwen2.5-VL-7B backbone, GAP achieves the best aggregate perception and reasoning scores over prior visual-latent baselines and our supervised variants. Controlled ablations show that PCA alignment, curated latent supervision, and difficulty-aware assignment each contribute to the gains, and inference-time interventions show that generated latents carry task-relevant visual signal beyond merely adding token slots.

3. **Curated latent-supervision data.** We construct a 49K high-quality multimodal latent-supervision dataset spanning visual CoT, charts, geometry, multimodal math, counting, and visual search. We release it through our code repository to support future work on visual latent reasoning.

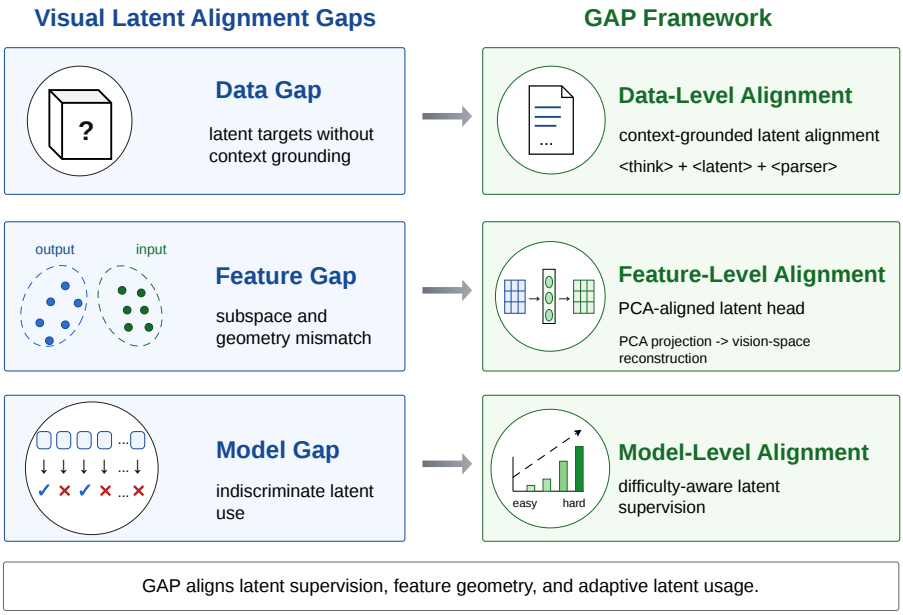


Figure 1: Conceptual overview of GAP. GAP addresses visual latent learning through three alignment levels: context-grounded latent supervision at the data level, a PCA-aligned latent head at the feature level, and difficulty-aware latent supervision at the model level.

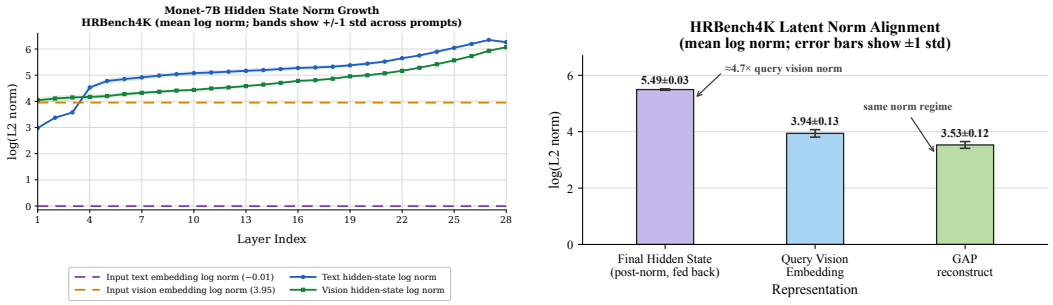
Beyond the high-level overview in Figure 1, Appendix Figure 3 traces the token-level execution path, showing how ordinary text decoding and GAP latent reconstruction interleave during generation.

2 Related Work

Explicit visual tools and generated visual thoughts. One way to supply missing visual evidence is to call external visual operations such as zooming, cropping, or region search. ZoomEye [16] builds a tree-based exploration strategy that enables MLLMs to inspect relevant image regions. Another line generates explicit visual thoughts or auxiliary images: Visualization-of-Thought [10], Latent Sketchpad [28], Render-of-Thought [23], DeepEyes [31], and GoT-R1 [4] use sketching, rendered reasoning traces, visual tools, or generation-oriented reinforcement learning. These methods provide interpretable intermediate artifacts, but they introduce additional tool or generation dependencies at inference. Our narrower goal is to keep inference inside the MLLM and to study how continuous visual feedback can be made compatible with the model's input-embedding distribution.

Continuous and visual latent reasoning. Latent reasoning represents intermediate computation as continuous embeddings rather than explicit text or images. Deng et al. [3] study the transition from explicit to implicit chain-of-thought, and COCONUT [7] trains language models to reason in a continuous latent space. In multimodal reasoning, Monet [21], LVR [9], Machine Mental Imagery [27], Chain-of-Visual-Thought [15], LIVR [11], VaLR [8], LaViT [24], and CrystaL [29] explore latent visual tokens, vision-aligned latent thoughts, or emergent visual latents. These works make visual latent reasoning an active and competitive direction; our contribution is therefore not the introduction of visual latent tokens themselves. Instead, we isolate a concrete feature-space failure mode of output-as-input latent feedback in pre-norm MLLMs and test a lightweight PCA-based reconstruction path that maps generated latents back toward the empirical vision-embedding subspace before autoregressive re-injection.

Residual norms and input compatibility in pre-norm transformers. Pre-norm transformers stabilize deep training by normalizing each sublayer input, but their residual streams can accumulate large norms across depth. Recent analyses and architectures such as SiameseNorm [12], Attention Residuals [20], and manifold-constrained hyper-connections [25] study the behavior of residual



(a) Layer-wise hidden-state norm growth.

(b) Latent norm alignment after GAP reconstruction.

Figure 2: Feature-space mismatch and GAP reconstruction on HRBench4K samples. (a) In Monet-7B, text and vision hidden-state log norms, measured before the final RMSNorm, grow sharply through the decoder stack and remain far above the input embedding regime; shaded bands show one standard deviation across prompts of $\log(\mathrm{L2\ norm})$ for hidden-state norms only. (b) The final decoder hidden state that output-as-input latent feedback reuses, taken after the final RMSNorm, is still about $4.7\times$ larger than the query vision-embedding norm, whereas GAP reconstruction maps generated latents back to the same norm regime before autoregressive re-injection.

streams and normalization. Our work connects this architectural issue to visual latent feedback. A vector that is suitable for output readout need not be distributed like an input vision embedding, even after the backbone's native final RMSNorm.

Selective latent computation. Latent and visual computation need not be applied uniformly to every example. Visual Enhanced Depth Scaling [6] studies when additional visual computation should be allocated during multimodal latent reasoning. Our difficulty-aware supervision is a training-time counterpart that decides which examples should include latent targets, without adding an inference-time routing or depth-scaling module.

3 Preliminary Analysis: Why Output-as-Input Latent Feedback Fails

Before presenting our method, we first establish the feature-space bottleneck behind output-as-input latent feedback in our Qwen2.5-VL-7B setting. Section 5.3 tests whether the latent head, curated data, difficulty-aware supervision, and PCA compression each matter, using component ablations under controlled variants.

3.1 Output-Embedding Norm Mismatch

Many current MLLM decoders, including Qwen2.5-VL, adopt a *pre-norm* architecture [26], where layer normalization is applied *before* each sub-layer (attention and FFN) rather than after. In pre-norm decoders, residual updates are added to an unnormalized stream, allowing hidden-state norms to accumulate with depth.

We profile the hidden-state norms of Monet-7B on HRBench4K samples (Figure 2). Monet feeds back the decoder state after the final RMSNorm as the next latent input. This post-norm state is still about $4.7\times$ the input vision-embedding norm at vision positions (Figure 2(b)) and about $290\times$ the input text-embedding norm at text positions. The final RMSNorm first normalizes each state to approximately unit RMS and then applies the learned elementwise gain γ ($\|\gamma\|_2 = 240.7$, much larger than the input-embedding norms). Before this norm, text and vision hidden-state norms grow steadily through the decoder stack (Figure 2(a)). The growth persists after Monet's latent fine-tuning, so Monet inherits it from its pre-norm Qwen2.5-VL-7B decoder. Let $\mathbf{h}^{(l)} \in \mathbb{R}^d$ denote the hidden state at layer l ($d = 3584$). The growth reflects residual accumulation in pre-norm architectures, where each transformer layer computes $\mathbf{h}^{(l+1)} = \mathbf{h}^{(l)} + \mathrm{SubLayer}(\mathrm{LN}(\mathbf{h}^{(l)}))$, where LN denotes layer normalization, SubLayer abstracts the layer's attention or feed-forward transformation, and the additive residual can accumulate norm across layers [12, 25, 20].

Existing latent reasoning methods [7, 21, 9] directly reuse the output hidden state $\mathbf{h}^{(L)}$ as the input embedding for the next latent token. This feeds a vector from a high-norm output-side distribution into a processing pipeline calibrated for the input-embedding norm regime, creating a distribution shift that destabilizes optimization. Thus, direct output-as-input reuse creates a feature-space mismatch in the latent feedback loop. For comparison, Figure 2(b) shows that GAP's reconstruction path returns latents to the query vision-embedding norm regime, while the post-norm final decoder hidden state that output-as-input reuse feeds back remains about $4.7\times$ larger.

Empirical evidence: norm rescaling alone helps. To validate that the norm gap is not merely a theoretical concern but a practical bottleneck, we conduct a simple experiment. We use Monet-7B as an output-as-input latent model, and compare it with and without a training-free EMA norm calibration that rescales the predicted latent to match the input vision embedding norm: $\tilde{\mathbf{v}}_t = \bar{n}_{\text{EMA}} \cdot \hat{\mathbf{v}}_t / \|\hat{\mathbf{v}}_t\|_2$, where $\bar{n}_{\text{EMA}}$ is the running average of $\|\mathbf{v}\|_2$ computed from the query image's vision embedding. As shown in Table 1, simply rescaling the norm at inference time already improves performance, confirming norm mismatch as a practical bottleneck.

Table 1: Effect of norm calibration. "Output-as-Input" denotes Monet-7B, a latent-reasoning model trained from Qwen2.5-VL-7B that reuses output-side hidden states as latent inputs. EMA norm calibration rescales the predicted latent to match input vision embedding norms at inference time, requiring no additional training. The improvement row is highlighted in green.

Configuration	**HRBench4K↑**	**MathVista↑**	**Avg.↑**
Output-as-Input (Monet-7B)	70.75	61.30	66.03
Output-as-Input + EMA norm calibration	71.63	63.30	67.46
Improvement	*+0.88*	*+2.00*	*+1.44*

The norm intervention motivates GAP's feature-level design, in which the model does not recycle raw decoder outputs as future visual inputs. We next describe the data-, feature-, and model-level alignment mechanisms, and then report controlled mechanism, data, PCA-rank, and token-budget ablations in Section 5.

4 Method

Guided by the alignment requirements in Section 1, GAP builds latent reasoning around three levels of alignment. Data-level alignment makes each continuous target traceable to a controllable auxiliary visual signal (§4.1); feature-level alignment maps generated latents back into the empirical vision-input subspace before feedback (§4.2); and model-level alignment applies latent supervision mainly to queries where the base model struggles (§4.3).

4.1 Data-Level Alignment: Controllable Latent Data Construction

Continuous latent supervision is otherwise hard to inspect. A latent vector can be supervised, but the example alone does not reveal what missing visual evidence that vector is meant to represent. We therefore construct 49,309 multimodal QA examples in which each query image and question is paired with an intermediate auxiliary image and a structured teacher response. During student training, the auxiliary image is not supplied as an image input to the student. Its frozen-ViT features fill the teacher-forced latent positions and also serve as targets for the latent head, while the surrounding `<think>` and `<parser>` text records the intended visual content. At inference time no auxiliary image is available, and the same latent positions are predicted autoregressively by GAP. The dataset is concentrated on visual CoT, chart understanding, GEOQA, multimodal math, counting, and visual search; full source composition and exact latent serialization are provided in Appendix K, with data-loader details in Appendix H. Auxiliary images are generated with Nano Banana 2 and structured responses are synthesized offline by Qwen3-VL-235B-A22B-Thinking; we estimate the construction cost at approximately US$3.31K in image-generation API fees and 274 GPU-hours for teacher synthesis (Appendix I).

Latent Interleave response format. Text reasoning → <latent> visual latent tokens </latent> → <parser> textual description of the intended latent visual evidence </parser> → continued text reasoning and final answer.

4.2 Feature-Level Alignment: PCA-Aligned Latent Head

At the feature level, the decoder state used to generate a latent token need not be distributed like the vision embeddings consumed as input. GAP avoids direct output-as-input reuse by predicting low-dimensional coordinates in an empirical auxiliary-image vision subspace and reconstructing each generated latent in the input vision-embedding coordinate system before re-injection. Let $\hat{\mathbf{v}}_{t-1} \in \mathbb{R}^d$ be the previous latent embedding fed to the transformer, $\mathbf{h}_t^{(L)}$ is the hidden state at the layer L, $\mathcal{M}_\theta$ the backbone, and $\mathcal{F}_\theta$ the latent head. Autoregressive latent generation is:

$$\mathbf{h}_t^{(L)} = \mathcal{M}_\theta(\hat{\mathbf{v}}_{t-1}, \mathrm{KV}_{<t}), \quad \mathbf{c}_t = \mathcal{F}_\theta(\bar{\mathbf{h}}_t), \quad \hat{\mathbf{v}}_t = \mathbf{P}_k \mathbf{c}_t + \boldsymbol{\mu}. \tag{1}$$

where $\bar{\mathbf{h}}_t = \mathrm{RMSNorm}(\mathbf{h}_t^{(L)})$, $\mathrm{KV}_{<t}$ denotes cached keys and values, and $\mathbf{c}_t \in \mathbb{R}^k$ are PCA coefficients. We estimate $\mathbf{P}_k$ and $\boldsymbol{\mu}$ once from training-only auxiliary-image embeddings; validation and benchmark embeddings are excluded. The basis is encoder-specific and frozen, and is reused across checkpoints, seeds, and tasks sharing the same encoder and fitting data. All experiments use 95% retained variance as a shared default, and Appendix M examines how the selected rank depends on the fitting distribution. PCA constrains the auxiliary latent pathway, while the original-image tokens remain accessible to the backbone. A full $\mathbb{R}^d \to \mathbb{R}^d$ latent MLP would require roughly $2d^2 \approx 25.7$M parameters for $d = 3584$; GAP instead uses $k = 629$ components, retaining 95% variance, and supervises the decoded embedding:

$$\mathcal{L}_{\text{latent}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left\| \mathbf{P}_k \mathcal{F}_\theta(\bar{\mathbf{h}}_t) + \boldsymbol{\mu} - \mathbf{v}_t \right\|_2^2, \tag{2}$$

where $\mathcal{T}$ is the set of latent token positions and $\mathbf{v}_t$ is the target auxiliary-image embedding. This reconstruction path is the feature-level alignment mechanism. Raw decoder states are never fed back directly, and Figure 2(b) shows that GAP reconstruction returns generated latents to the query vision-embedding norm regime. The full objective combines language modeling with latent alignment:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{latent}} \mathcal{L}_{\text{latent}}, \tag{3}$$

Appendix Figure 3 illustrates the text-token versus latent-token inference path, and Appendix H gives the deferred implementation details, including projector initialization and scheduled sampling ablations.

4.3 Model-Level Alignment: Difficulty-Aware Latent Supervision

Not all queries benefit from latent reasoning. The matched comparison in Section 5.3 shows that restricting latent supervision to examples the base model cannot solve improves all seven reported metrics. For each training query, we run $N = 8$ inference passes with the base Qwen2.5-VL 7B model, compute empirical accuracy $\hat{a}$, and assign supervision by:

$$\mathrm{DA}(q) = \begin{cases} \text{text-only} & \text{if } \hat{a} > \tau, \\ \text{latent} & \text{if } \hat{a} \leq \tau, \end{cases} \tag{4}$$

In the current experiments $\tau = 0$, so only examples the base model never solves across the eight samples receive latent supervision. For "text-only" queries, the model is trained with a standard text response; for "latent" queries, it uses the full latent format from Section 4.1. This model-level alignment teaches when latent-assisted reasoning is useful without adding an inference-time router, and the exact data-loader stripping rule is deferred to Appendix H.

5 Experiments

5.1 Experimental Setup

Base model and baselines. We use Qwen2.5-VL 7B [2] as the base MLLM. We compare against the zero-shot base model, a **Dense Caption SFT** control trained on the same 49K curated data

without latent tokens or auxiliary-image alignment, Monet [21], and LVR [9]. Our variants are a basic latent-head SFT model (LH), a difficulty-aware full latent head without PCA (LH+DA), an all-latent PCA-aligned latent-head model (LH+PCA), and the full difficulty-aware PCA-aligned version, GAP (LH+PCA+DA). Monet and the original LVR are prior-system baselines with their own training recipes and data. We additionally retrain LVR on the same 49K corpus (LVR-49K) to control the backbone and data; matched component ablations then examine PCA and difficulty-aware assignment within our recipe.

Benchmarks and metrics. Following the evaluation of Monet and LVR, we evaluate perception on HRBench4K, MMStar, and MME-RealWorld-Lite, and additionally evaluate reasoning on MathVista and WeMath. For HRBench4K, we report Overall, fine-grained spatial perception (FSP), and fine-grained counting perception (FCP). For MME-RealWorld-Lite, we report Overall, Reasoning, and Perception sub-scores. We define Avg-P as the average of HRBench4K Overall, MMStar, and MME-RealWorld-Lite Overall, and Avg-R as the average of MathVista, WeMath-S, and WeMath-L.

Implementation details. We fine-tune Qwen2.5-VL 7B using AdamW with a base learning rate of 1×10^{-5}, a latent-head learning rate of 1×10^{-5}, a global batch size of 128, and 2 epochs on the 33,279-example training split of the 49K curated dataset. Dense Caption SFT, LVR-49K, and all GAP variants use this same training split. The PCA basis is computed offline from training-set vision embeddings with a 95% explained-variance target, yielding $k = 629$ principal components. The latent loss weight is $\lambda_{\text{latent}} = 1.0$. The main difficulty-aware run uses 36 latent tokens; Appendix J reports the token-budget sweep, and Appendix I reports wall-clock inference cost. We use one shared budget of 36 tokens across benchmarks because it gives the best mean over HRBench4K, MMStar, and MathVista in our sweep, although 16 tokens is close and can be more efficient. Token count controls a perception–reasoning and accuracy–cost trade-off; the 144-token MME analysis in Appendix M is a separate capacity comparison, not a benchmark-specific replacement of the default model. Appendix Table 11 reports three-evaluator-seed mean $\pm$ standard deviation scores for each token budget.

5.2 Main Results

Tables 2 and 3 evaluate the overall effectiveness of GAP. We first ask whether the three-level alignment recipe improves both perception and reasoning.

Comparison with prior latent methods. GAP (LH+PCA+DA) denotes the full method with a latent head (LH), PCA-aligned latent reconstruction (PCA), and difficulty-aware latent assignment (DA). The prior latent baselines improve perception but lose reasoning accuracy relative to the zero-shot Qwen2.5-VL 7B baseline: Avg-R drops from 52.62 to 47.99 for Monet and 47.67 for LVR, matching the norm-mismatch failure mode analyzed in Section 3.1. Monet and LVR use their own training recipes and data, so these rows are prior-system comparisons; the LVR-49K control below matches the data. Under this comparison, Ours-GAP (LH+PCA+DA) improves Avg-P by **+1.89** and Avg-R by **+5.98** over Monet, and improves Avg-P by **+0.71** and Avg-R by **+6.30** over LVR.

Separating data and method effects. Retraining LVR on the 49K corpus improves Avg-P from 60.76 to 60.86 and Avg-R from 47.67 to 49.69, confirming a contribution from the curated data. With the backbone and data fixed, GAP further improves these aggregates by +0.61 and +4.28 over LVR-49K. This comparison separates the data change from the remaining method differences; Appendix Table 12 summarizes it together with the matched component comparisons that isolate latent supervision, PCA alignment, and DA. Gains on individual benchmarks are not uniform. On MME-RW-Lite, LVR-49K and the default GAP both trail the original LVR on Overall; we attribute this to the limited coverage of real-world scenarios in our training mixture, and a larger latent budget recovers most of the gap (Appendix M.4).

Ours-GAP (LH+PCA+DA) achieves the highest Avg-P and Avg-R of all methods and is the only latent method that improves both aggregates over the zero-shot base model.

Takeaway 1. The three-level alignment recipe gives the best aggregate perception and reasoning performance and is the only latent method that improves both over the base model.

Why dense captioning helps perception but hurts reasoning. Dense Caption SFT improves the perception aggregate from 57.66 to 59.20, because more detailed textual descriptions of the image help tasks that primarily require fine-grained visual recognition. However, it still trails the base

Table 2: Main perception results. HRBench4K reports Overall, FSP, and FCP; MMStar reports Overall; MME-RealWorld-Lite reports Overall, Reasoning, and Perception. Avg-P averages HRBench4K Overall, MMStar Overall, and MME-RealWorld-Lite Overall. Best results are in **green** and second-best in yellow.

Method	HRBench4K			MMStar	MME-RW-Lite			Avg-P
	Ovr.↑	FSP↑	FCP↑	Ovr.↑	Ovr.↑	Rsn.↑	Perc.↑	
Qwen2.5-VL	68.13	83.50	52.75	60.46	44.39	37.60	48.76	57.66
Dense Cap	69.38	85.50	53.25	60.50	47.73	42.40	51.15	59.20
Monet-7B	70.75	**90.50**	51.00	60.80	47.20	42.90	50.00	59.58
LVR	70.75	84.25	57.25	62.07	**49.45**	43.07	**53.55**	60.76
LVR-49K	71.63	84.50	58.75	62.17	48.78	**43.20**	52.36	60.86
Ours-LH	71.38	85.50	57.25	61.20	45.67	39.20	49.82	59.42
Ours-LH+DA	71.88	85.00	58.75	61.80	45.75	39.06	50.00	59.81
Ours-LH+PCA	72.25	87.75	56.75	62.10	46.93	39.80	51.50	60.43
Ours-GAP (LH+PCA+DA)	**73.25**	86.50	**60.00**	**63.40**	47.77	43.07	50.79	**61.47**

Table 3: Main reasoning results. MathVista measures multimodal mathematical reasoning, and WeMath is reported with strict (S) and loose (L) accuracy. Avg-R averages MathVista, WeMath-S, and WeMath-L. Best results are in **green** and second-best in yellow.

Method	MathVista↑	WeMath-S↑	WeMath-L↑	Avg-R↑
Qwen2.5-VL 7B	68.50	36.29	53.06	52.62
Dense Cap SFT	67.70	28.60	45.43	47.24
Monet-7B	61.30	32.67	50.00	47.99
LVR	67.00	26.95	49.05	47.67
LVR-49K	68.82	30.59	49.65	49.69
Ours (LH)	68.60	30.00	48.86	49.15
Ours (LH+DA, no PCA)	68.90	31.26	50.00	50.05
Ours (LH+PCA)	70.20	35.24	52.00	52.48
Ours-GAP (LH+PCA+DA)	**71.00**	**36.33**	**54.57**	**53.97**

model on MathVista and substantially degrades WeMath, reducing the reasoning aggregate from 52.62 to 47.24. We attribute this to a mismatch between dense descriptive supervision and multi-hop reasoning.

Why visual latents need alignment and selectivity. The basic LH row improves over Dense Caption SFT on HRBench4K, MathVista, and both aggregates (+0.22 Avg-P, +1.91 Avg-R), but it still trails the base model on Avg-R (49.15 versus 52.62), so a latent head alone is not sufficient for robust aggregate gains. PCA alignment addresses this by reconstructing generated latents in the empirical subspace spanned by real training-set vision embeddings, bringing the fed-back latents closer to the input vision-embedding subspace and norm regime. It also reduces the latent-head dimensionality, lowering the risk that the full head overfits the curated latent-supervision distribution; on the OOD POPE check in Appendix G, adding PCA restores the overall score from 85.72 to 86.32, on par with the base model (86.23). DA retains text supervision on examples the base model already solves and reserves latent targets for harder cases; the matched comparison in Section 5.3 isolates its effect. Neither partial variant exceeds the zero-shot base on Avg-R (50.05 for LH+DA and 52.48 for LH+PCA, versus 52.62), whereas combining both components reaches 53.97. The two partial variants expose complementary failure modes rather than instability of the full recipe. LH+DA leaves the output–input feature mismatch unresolved, whereas LH+PCA imposes *overly strong* latent supervision by forcing latent targets onto every example, including those the base model already solves. The full method gains +1.35 Avg-R over the base model, and this gain requires both components.

5.3 Component Ablations

We next separate the remaining ingredients behind GAP beyond the basic LH row in Tables 2 and 3: curated latent data with difficulty-aware supervision and PCA-aligned latent generation. These ablations complement the full benchmark suite above by isolating narrower two- or three-benchmark comparisons.

Data quality and difficulty-aware supervision. Table 4 compares latent-supervision variants that differ in data source and assignment strategy. The curated 49K latent mixture improves substantially over training our latent-head recipe on Monet's 125K SFT data, especially on MathVista. Without PCA, DA adds only +0.40 over all-latent supervision, since both variants still suffer from the unresolved feature mismatch. We therefore isolate DA inside the PCA-aligned recipe, comparing LH+PCA with LH+PCA+DA under the same training configuration and changing only the supervision assignment.

Table 4: Component comparison for latent supervision data and assignment strategy. Best results are in **green** and second-best in yellow.

Training Data / Strategy	Size	HRBench4K↑	MathVista↑	Avg.↑
Monet latent model	125K	70.75	61.30	66.03
Ours (latent head on Monet SFT data)	125K	71.00	62.70	66.85
Ours, all latent	49K	71.38	68.60	69.99
Ours, difficulty-aware (LH+DA)	49K	**71.88**	**68.90**	**70.39**

Takeaway 2. Within the PCA-aligned recipe, difficulty-aware assignment improves both aggregates in the main tables (+1.04 Avg-P, +1.49 Avg-R) and all seven metrics in a five-seed matched evaluation.

Matched difficulty-aware comparison. We evaluate fixed prediction sets from LH+PCA and LH+PCA+DA across five evaluator seeds (Appendix Table 13). DA increases HRBench4K Overall from 72.25 ± 0.1581 to 73.53 ± 0.1559 and MathVista from 70.06 ± 0.2348 to 71.20 ± 0.1643, and raises the mean on all seven reported metrics. The two models differ only in which examples receive latent targets, so these gains come from withholding latent supervision on examples the base model already solves, which removes the overly strong supervision of LH+PCA.

PCA components and capacity control. Table 5 varies the number of retained PCA components while keeping the backbone, curated training data, 36 latent tokens, difficulty-aware supervision, and evaluation benchmarks fixed. The uncompressed LH+DA head already improves over the base model, but all PCA+DA variants further improve Avg-3 while predicting only 7–18% as many output dimensions, so the input-space PCA parameterization helps beyond head capacity. The default 95% setting gives the best three-benchmark mean; Appendix M examines its sensitivity to the fitting distribution and shows that GAP improves the MathVista geometry subset, where fine geometric detail matters, by +5.39 points over the base model.

Table 5: PCA component-count and capacity-control ablation under difficulty-aware latent supervision. Avg-3 averages HRBench4K Overall, MMStar, and MathVista. Δ is measured relative to Qwen2.5-VL-7B Avg-3. Best results are in **green** and second-best in yellow.

Method	# Comp.	RelMSE↓	HR4K	MMStar	MVista	Avg-3	Δ
Qwen2.5-VL	–	–	68.13	60.46	68.50	65.70	0.00
LH+DA	3584	–	71.88	61.80	68.90	67.53	+1.83
PCA-0.85+DA	265	0.1494	71.00	62.60	69.30	67.63	+1.93
PCA-0.90+DA	381	0.0999	71.38	61.60	70.10	67.69	+1.99
GAP (PCA-0.95+DA)	629	**0.0500**	**73.25**	**63.40**	**71.00**	**69.22**	**+3.52**

Takeaway 3. PCA-aligned reconstruction outperforms an unconstrained latent head at every tested rank.

PCA-aligned reconstruction thus acts as an effective capacity-and-subspace constraint; comparisons with other low-rank parameterizations are left to future work.

5.4 Generated Latents Contain Task-Relevant Visual Signal

To test whether gains come from content-bearing latents rather than formatting or training-time regularization, we evaluate inference-time interventions on HRBench4K and MathVista. The zero-latent checkpoint refers to the GAP checkpoint evaluated with latent-token generation suppressed, so no generated PCA-decoded latent embedding is re-injected into the input stream; this row measures training and data effects without inference-time latent feedback. The noise setting replaces predicted PCA coefficients with Gaussian noise before reconstruction. The clean and noise settings use the same reconstruction and norm-matching rules.

Table 6: Latent content intervention. The zero-latent checkpoint row uses GAP with latent-token generation suppressed, while the noise row replaces predicted PCA coefficients with Gaussian noise before reconstruction. Avg-2 averages HRBench4K Overall and MathVista; Δ is relative to Qwen2.5-VL-7B Avg-2. Best results are in **green** and second-best in yellow.

Intervention	Ckpt.	Latents	Content	HR4K↑	MVista↑	Avg-2↑	Δ
Qwen2.5-VL	Base	0	None	68.13	68.50	68.32	0.00
Dense Cap	Dense	0	None	69.38	67.70	68.54	+0.22
Zero-latent ckpt.	GAP	0	None	71.25	69.40	70.33	+2.01
Noise latents	GAP	36	Gauss. noise	70.38	69.00	69.69	+1.37
Clean latents	GAP	36	Pred. visual	**73.25**	**71.00**	**72.13**	**+3.81**

Takeaway 4. Clean generated latents provide useful inference-time visual signal.

The zero-latent checkpoint still beats Dense Caption SFT by +1.79 Avg-2, so latent training improves the model even when latent feedback is disabled at inference. Using clean generated latents further improves Avg-2 from 70.33 to 72.13, while Gaussian coefficient noise drops below both the zero-latent checkpoint and clean-latent setting. Thus, clean predicted latents carry signal beyond formatting and training-time effects.

The main setting uses 36 latent tokens; Appendix J reports the full token-budget sweep. Visual latent capacity is therefore most effective when it is aligned, content-bearing, and budgeted rather than simply increased.

6 Conclusion

Reusing decoder outputs as visual latents feeds a pre-norm MLLM with inputs it was not trained to consume, because even after the final RMSNorm the fed-back state is about $4.7\times$ the norm of the input vision embeddings. GAP removes this mismatch by aligning visual latents at three levels, namely context-grounded latent supervision at the data level, PCA-aligned reconstruction at the feature level, and difficulty-aware latent assignment at the model level. On Qwen2.5-VL-7B, GAP is the only latent method in our comparison that improves both perception and reasoning over the base model (+3.81 Avg-P and +1.35 Avg-R), whereas Monet and LVR gain perception but lose 4.6 and 5.0 points of Avg-R, respectively. The levels are complementary, since neither feature alignment nor difficulty-aware assignment alone surpasses the base model on reasoning, while their combination does, and inference-time interventions show that the generated latents carry task-relevant visual signal. These results suggest that visual latents should be treated as model inputs that must match the input-embedding distribution, not as raw outputs to recycle. Our evaluation covers one backbone and five benchmarks with an encoder-specific PCA basis and a shared 95% retained-variance default, and Appendix B discusses the remaining limitations, including the latent-token budget trade-off.

Acknowledgments and Disclosure of Funding

This work was conducted during Yanting Miao's internship at Alibaba Group. We thank Alibaba Group for its support of this research. We also thank Li Xu and Yudong Luo for their feedback on the writing and for early discussions of the idea. Poupart also acknowledges funding from the Canada CIFAR AI Chair program, the Natural Sciences and Engineering Research Council of Canada and a grant from IITP & MSIT of Korea (No. RS2024-00457882, AI Research Hub Project).

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL `https://arxiv.org/abs/2511.21631`.

[2] 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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

[3] Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

[4] Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. *arXiv preprint arXiv:2505.17022*, 2025.

[5] Google DeepMind. Gemma 4 model card. `https://ai.google.dev/gemma/docs/core/model_card_4`, 2026. Accessed: 2026-05-06.

[6] Yudong Han, Yong Wang, Zaiquan Yang, Zhen Qu, Liyuan Pan, and Xiangxiang Chu. Visual enhanced depth scaling for multimodal latent reasoning. *arXiv preprint arXiv:2604.10500*, 2026.

[7] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

[8] Byungwoo Jeon, Yoonwoo Jeong, Hyunseok Lee, Minsu Cho, and Jinwoo Shin. Vision-aligned latent reasoning for multi-modal large language model. *arXiv preprint arXiv:2602.04476*, 2026.

[9] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025.

[10] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

[11] Kelvin Li, Chuyi Shang, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, and Roei Herzig. Latent implicit visual reasoning. *arXiv preprint arXiv:2512.21218*, 2025.

[12] Tianyu Li, Dongchen Han, Zixuan Cao, Haofeng Huang, Mengyu Zhou, Ming Chen, Erchao Zhao, Xiaoxi Jiang, Guanjun Jiang, and Gao Huang. Siamesenorm: Breaking the barrier to reconciling pre/post-norm. *arXiv preprint arXiv:2602.08064*, 2026.

[13] Chengzhi Liu, Yuzhe Yang, Yue Fan, Qingyue Wei, Sheng Liu, and Xin Eric Wang. Reasoning within the mind: Dynamic multimodal interleaving in latent space. *arXiv preprint arXiv:2512.12623*, 2025.

[14] Yanting Miao, William Loh, Suraj Kothawade, Pascal Poupart, Abdullah Rashwan, and Yeqing Li. Subject-driven text-to-image generation via preference-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 123563–123591, 2024.

[15] Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and XuDong Wang. Chain-of-visual-thought: Teaching vlms to see and think better with continuous visual tokens. *arXiv preprint arXiv:2511.19418*, 2025.

[16] Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6613–6629, 2025.

[17] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[18] Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025.

[19] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[20] Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, et al. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026.

[21] Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025.

[22] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[23] Yifan Wang, Shiyu Li, Peiming Li, Xiaochen Yang, Yang Tang, and Zheng Wei. Render-of-thought: Rendering textual chain-of-thought as images for visual latent reasoning. *arXiv preprint arXiv:2601.14750*, 2026.

[24] Linquan Wu, Tianxiang Jiang, Yifei Dong, Haoyu Yang, Fengji Zhang, Shichaang Meng, Ai Xuan, Linqi Song, and Jacky Keung. Lavit: Aligning latent visual thoughts for multi-modal reasoning. *arXiv preprint arXiv:2601.10129*, 2026.

[25] Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, et al. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

[26] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

[27] Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

[28] Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, et al. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in mllms. *arXiv preprint arXiv:2510.24514*, 2025.

[29] Yang Zhang, Danyang Li, Yuxuan Li, Xin Zhang, Tianyu Xie, Mingming Cheng, and Xiang Li. Crystal: Spontaneous emergence of visual latents in mllms. *arXiv preprint arXiv:2602.20980*, 2026.

[30] Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025.

[31] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

A Architecture Illustration

Figure 3 makes the interleaving explicit: ordinary text tokens are decoded through the LM head, latent spans are routed through the PCA-aligned latent head, and each reconstructed visual embedding is fed back as the next latent-token input.

B Limitations and Scope

1. The full training and evaluation pipeline is validated on Qwen2.5-VL 7B and Monet-7B. Extending the same analysis to other pre-norm MLLMs is necessary before claiming cross-backbone universality. Post-norm backbones may still suffer from latent subspace mismatch, but the norm-growth mechanism studied here is specific to pre-norm residual streams. Our benchmark suite follows the perception benchmarks of Monet and LVR and adds MathVista and WeMath for reasoning; broader task coverage remains to be tested.
2. The LVR-49K control matches the training data and backbone to GAP, while Monet remains a prior-system baseline. The matched-data comparison helps separate the contribution of the curated corpus from that of the overall training method, but it does not isolate PCA or difficulty-aware assignment individually; those effects require the corresponding component ablations.
3. The difficulty threshold $\tau = 0$ is heuristic. The matched LH+PCA versus LH+PCA+DA control holds the training configuration fixed and varies the supervision assignment. Its five evaluator-seed repeats measure evaluation variability on fixed prediction sets, not variability across independently trained models, and therefore do not by themselves establish training-seed robustness of the DA effect. Sensitivity to other difficulty thresholds remains to be studied.
4. PCA constrains latents to an empirical affine subspace and reduces head capacity. Appendix F adds a component-count sweep, but random-basis, no-mean, and matched-parameter full-head controls remain incomplete. The numerical basis is encoder-specific and must be re-estimated for a new vision encoder or fitting distribution, and the rank selected at a given retained-variance ratio depends on the fitting distribution (Appendix M.3). All experiments use 95% as a shared default. Variance retention also does not guarantee that low-variance but task-relevant directions are preserved.
5. Gains are not uniform across benchmarks. With the shared 36-token budget, GAP trails the original LVR on MME-RealWorld-Lite, whose remote-sensing, monitoring, and in-the-wild OCR scenarios are underrepresented in our chart-, geometry-, and visual-search-oriented training mixture; LVR-49K shows a similar decrease. Increasing the budget to 144 tokens recovers most of this gap (Appendix M.4), but no single fixed budget is best on every benchmark, so the token count should be treated as a perception–reasoning and accuracy–cost trade-off.
6. The `<parser>` field describes the intended auxiliary visual evidence; it is not a decoding of the continuous latent representation or a guarantee that the latent vector faithfully encodes that description. Stronger parser–latent faithfulness tests are left for future work.
7. The auxiliary-image requirement limits the approach to domains where meaningful auxiliary views can be generated or sourced and requires careful duplicate and leakage checks against evaluation benchmarks. We have not systematically repeated the full data-construction pipeline with alternative teachers or image generators, so sensitivity to these choices remains unquantified.
8. Future work should test the same alignment design on additional MLLM backbones and study whether RL-based latent training gives further gains once the feedback space is input-compatible. Our implementation uses a fixed latent-token budget, as in Monet and LVR; allocating latent tokens dynamically per example, e.g., based on task difficulty, image complexity, or generation confidence, is another open direction.

C Broader Impact

GAP aims to make visual latent reasoning more reliable and inspectable by aligning generated latent tokens with input-compatible visual features and by pairing latent supervision with a human-readable parser trace. This may benefit multimodal systems that need compact intermediate visual reasoning without calling external visual tools or image generators at inference time, potentially reducing pipeline complexity and inference-time resource use.

At the same time, stronger visual reasoning systems can also make incorrect answers appear more grounded. The <parser> field describes the intended visual evidence, but it is not a formal guarantee that the continuous latent vector faithfully encodes exactly that content. We therefore view parser traces, intervention tests, and qualitative examples as diagnostic aids rather than complete explanations. Future deployments should include task-specific validation, leakage checks, and caution in high-stakes settings where visual errors could affect safety, medical, legal, or financial decisions.

D External Assets, Licenses, and Terms

Table 7 summarizes the existing models, datasets, benchmarks, services, and comparison assets used in this work. License and terms information is taken from the upstream pages listed in the table. When an upstream benchmark does not declare a standard license, or declares restrictive research-only terms, we use it only for evaluation or comparison and do not redistribute the benchmark content. The released latent-supervision data (36,985 examples; Appendix K) records the upstream source of every example in a per-example `data_source` field, and its dataset card maps each source to the upstream page and license listed in Table 7. The release is intended for non-commercial research, and each example remains subject to the license and terms of its upstream source.

Table 7: External assets, upstream pages, and license or terms-of-use status.

Asset	Role in this work	Upstream page or version	License / terms status and use
Qwen2.5-VL-7B-Instruct	Student backbone and zero-shot baseline	`https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct`	Apache-2.0. We use the model as the base MLLM and preserve upstream attribution in released model metadata.
Qwen3-VL-235B-A22B-Thinking	Offline teacher for response synthesis	`https://huggingface.co/Qwen/Qwen3-VL-235B-A22B-Thinking`	Apache-2.0. Used only to synthesize structured supervision; no teacher weights are redistributed.
Gemini image-generation service (Nano Banana 2)	Auxiliary-image generator during data construction	`https://ai.google.dev/gemini-api/terms`	Governed by Gemini API / Google AI Studio terms and the associated prohibited-use policy. We use generated auxiliary images for research supervision and do not redistribute the service or model weights.
Monet-7B and Monet-SFT-125K	Prior latent baseline, data-source comparison, and source of our Visual-CoT, Zebra-CoT, CogCoM, and ReFocus training examples	`https://huggingface.co/NOVAglow646/Monet-7B`; `https://huggingface.co/datasets/NOVAglow646/Monet-SFT-125K`	Hugging Face model and dataset pages list MIT. We cite the Monet paper and use Monet-7B for baseline comparison. We use Monet-SFT-125K for the data ablation in Table 4 and as the source of our Visual-CoT, Zebra-CoT, CogCoM, and ReFocus examples, which remain subject to the original dataset terms listed below.
LVR	Prior latent-method comparison	`https://arxiv.org/abs/2511.19094`	Paper comparison only; we do not redistribute LVR code or model weights. If released upstream assets are used in future versions, their license will be added to this table.
HR-Bench / HRBench4K	Perception evaluation benchmark	`https://huggingface.co/datasets/DreamMr/HR-Bench`	Upstream Hugging Face metadata lists license as `other`, and the repository license file is empty. Used only for evaluation; benchmark content is not redistributed.
MMStar	Perception evaluation benchmark	`https://github.com/MMStar-Benchmark/MMStar`; `https://huggingface.co/datasets/Lin-Chen/MMStar`	No standard dataset license is declared on the primary GitHub / Hugging Face pages. Used only for evaluation; benchmark content is not redistributed.
MME-RealWorld-Lite	Perception and reasoning evaluation benchmark	`https://github.com/MME-Benchmarks/MME-RealWorld`; `https://huggingface.co/datasets/yifanzhang114/MME-RealWorld-Lite`	Upstream terms restrict use to academic research and prohibit commercial use and redistribution without approval. We use the Lite split only for evaluation and do not redistribute it.
MathVista	Reasoning evaluation benchmark and qualitative example source	`https://huggingface.co/datasets/AI4Math/MathVista`	CC-BY-SA-4.0 for new dataset contributions, with source-image copyrights retained by original owners and dataset-specific test-set / no-training conditions. We use it only for evaluation and qualitative analysis.
We-Math	Reasoning evaluation benchmark	`https://huggingface.co/datasets/We-Math/We-Math`	CC-BY-NC-4.0. Used only for evaluation; benchmark content is not redistributed.

Continued on next page

Asset	Role in this work	Upstream page or version	License / terms status and use
POPE	OOD hallucination-sensitive sanity check	https://github.com/RUCAIBox/POPE	MIT license on the official repository. Used only for evaluation in Appendix G.
Visual-CoT	Source pool for visual-CoT-style training examples	https://huggingface.co/datasets/deepcs233/Visual-CoT	Hugging Face metadata lists Apache-2.0, while the dataset card also states Attribution-NonCommercial 4.0 and research-purpose restrictions. Released examples keep these terms and are treated as CC BY-NC 4.0.
Zebra-CoT	Source pool for counting, visual-search, and geometry examples	https://huggingface.co/datasets/multimodal-reasoning-lab/Zebra-CoT	CC-BY-NC-4.0. Used for research training data construction; redistribution follows the upstream non-commercial terms.
CogCoM / CoMDataset	Focused visual reasoning source pool	https://huggingface.co/datasets/qijimrc/CoMDataset; https://github.com/THUDM/CogCoM	Dataset contributions are CC-BY-SA-4.0 with original source copyrights retained; CogCoM code is Apache-2.0 and model weights have a separate model license. We use the dataset portion only where terms permit.
ReFocus	Structured visual reasoning source pool	https://huggingface.co/datasets/ReFocus/ReFocus_Data	Apache-2.0. Used for research training data construction with upstream attribution.
MathVision	Multimodal math source pool	https://huggingface.co/datasets/MathLLMs/MathVision	MIT. Used for multimodal math examples in the training mixture with upstream attribution.
GeoQA and Geometry3K / Geo3K	Geometry source pools	https://huggingface.co/datasets/hz2475/geoQA; https://huggingface.co/datasets/xyliu6/geometry3k	The referenced GeoQA port lists MIT; the referenced Geometry3K port lists Apache-2.0. We include source identifiers in the release manifest and avoid exact evaluation duplicates as described in Appendix K.
ChartQA	Chart-understanding source pool	https://huggingface.co/datasets/ahmed-masry/ChartQA; https://github.com/vis-nlp/ChartQA	GPL-3.0 on the Hugging Face and GitHub pages; chart images were crawled by the ChartQA authors from online sources such as Statista and Pew Research Center, and their copyrights are retained by the original owners. Used for research training data construction with upstream attribution.
MM-MATH	Multimodal math source pool	https://huggingface.co/datasets/THU-KEG/MM_Math	MIT. Problems are collected from Chinese middle-school exams. Used for multimodal math examples in the training mixture with upstream attribution.

E Background: PCA and Pre-Norm Decoders

PCA reconstruction. Let $\{\mathbf{v}_i\}_{i=1}^N$ denote vision embeddings extracted from auxiliary images by the frozen vision encoder, where $\mathbf{v}_i \in \mathbb{R}^d$. We compute the empirical mean and covariance

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i, \qquad \boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{v}_i - \boldsymbol{\mu})(\mathbf{v}_i - \boldsymbol{\mu})^\top. \tag{5}$$

Let $\mathbf{P}_k = [\mathbf{p}_1, \ldots, \mathbf{p}_k] \in \mathbb{R}^{d\times k}$ be the top-k eigenvectors of $\boldsymbol{\Sigma}$. For any centered vision embedding, PCA gives coefficients and a rank-k reconstruction:

$$\mathbf{c} = \mathbf{P}_k^\top(\mathbf{v} - \boldsymbol{\mu}), \qquad \hat{\mathbf{v}} = \mathbf{P}_k\mathbf{c} + \boldsymbol{\mu}. \tag{6}$$

We report the relative reconstruction error as the fraction of centered vision-embedding variance not captured by the rank-k subspace:

$$\mathrm{RelMSE}(k) = \frac{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{v}_i - \left(\mathbf{P}_k\mathbf{P}_k^\top(\mathbf{v}_i - \boldsymbol{\mu}) + \boldsymbol{\mu}\right)\right\|_2^2}{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{v}_i - \boldsymbol{\mu}\right\|_2^2} = 1 - \frac{\sum_{j=1}^{k}\lambda_j}{\sum_{j=1}^{d}\lambda_j}, \tag{7}$$

where λ_j are the eigenvalues of the empirical covariance. Lower relative reconstruction error means that the PCA subspace retains more of the auxiliary-image vision-embedding variance. In our model, the latent head predicts $\mathbf{c}_t$ rather than an unconstrained full-dimensional embedding. The generated latent is then reconstructed by Eq. 6, which constrains feedback tokens to the affine subspace estimated from real vision embeddings.

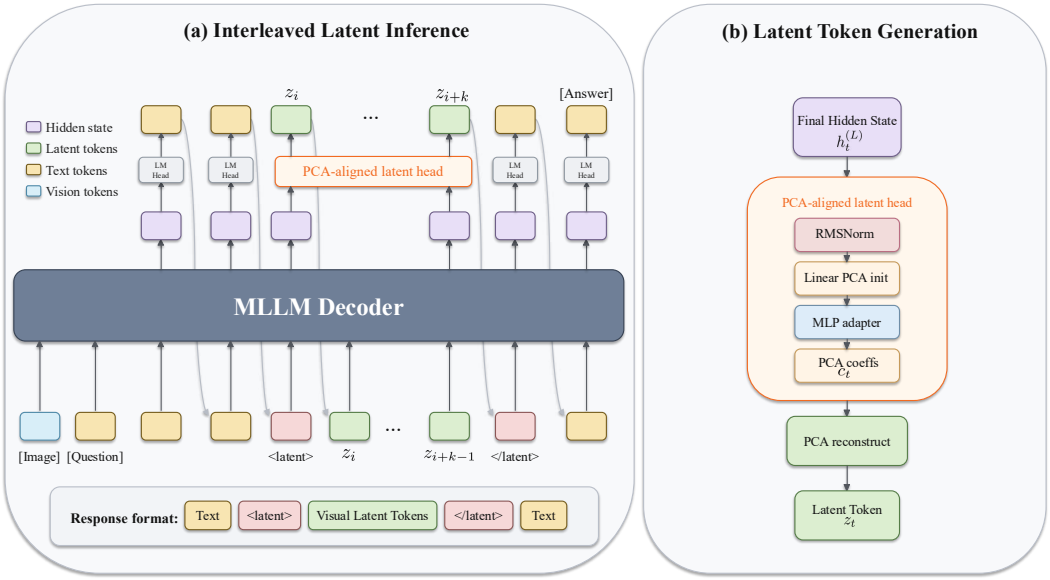


Figure 3: Interleaved visual latent inference and latent-token generation. (a) The MLLM generates ordinary text tokens through the LM head and generates visual latent tokens inside a `<latent>` span. Each generated latent token is fed back autoregressively as a subsequent latent-token input, giving a response format of text, visual latent tokens, and text. (b) A latent token is generated from the final decoder state by applying the backbone's native final RMSNorm, a PCA-aligned latent head that predicts PCA coefficients, and a fixed PCA reconstruction step. The reconstructed embedding is then used as the next latent token; auxiliary-image targets are used only to train this path.

Pre-norm decoder states. Modern decoder-only transformers, including the Qwen2.5-VL backbone used in our experiments, are typically pre-norm architectures. Ignoring layer indices on attention keys and values for readability, a pre-norm block has the form

$$\mathbf{x}'_\ell = \mathbf{x}_\ell + \mathrm{Attn}_\ell(\mathrm{RMSNorm}_\ell(\mathbf{x}_\ell)) \,, \tag{8}$$

$$\mathbf{x}_{\ell+1} = \mathbf{x}'_\ell + \mathrm{MLP}_\ell\big(\mathrm{RMSNorm}'_\ell(\mathbf{x}'_\ell)\big) \,. \tag{9}$$

RMSNorm itself rescales a hidden vector by its root-mean-square magnitude:

$$\mathrm{RMSNorm}(\mathbf{x}) = \mathbf{g} \odot \frac{\mathbf{x}}{\sqrt{\frac{1}{d}\sum_{j=1}^{d} x_j^2 + \epsilon}}, \tag{10}$$

where $\mathbf{g}$ is a learned scale vector. This placement explains why hidden-state norms can accumulate with depth. For a single residual update, write

$$\mathbf{x}_{\ell+1} = \mathbf{x}_\ell + \mathbf{u}_\ell, \qquad \mathbf{u}_\ell = F_\ell(\mathrm{RMSNorm}(\mathbf{x}_\ell)) \,, \tag{11}$$

where F_ℓ denotes the attention or MLP branch. Then

$$\|\mathbf{x}_{\ell+1}\|_2^2 = \|\mathbf{x}_\ell\|_2^2 + \|\mathbf{u}_\ell\|_2^2 + 2\,\langle \mathbf{x}_\ell, \mathbf{u}_\ell \rangle \,. \tag{12}$$

Pre-norm controls the input scale seen by F_ℓ, but it does not normalize the residual stream after $\mathbf{u}_\ell$ is added. Therefore, if residual updates have nonzero energy and are not consistently anti-aligned with the current residual stream, i.e.,

$$\mathbb{E}[\langle \mathbf{x}_\ell, \mathbf{u}_\ell \rangle] \approx 0, \qquad \mathbb{E}\Big[\|\mathbf{u}_\ell\|_2^2\Big] > 0, \tag{13}$$

then the expected squared norm accumulates:

$$\mathbb{E}\Big[\|\mathbf{x}_L\|_2^2\Big] \approx \mathbb{E}\Big[\|\mathbf{x}_0\|_2^2\Big] + \sum_{\ell=0}^{L-1} \mathbb{E}\Big[\|\mathbf{u}_\ell\|_2^2\Big] \,. \tag{14}$$

This is not a monotonicity guarantee for every individual layer, because the inner-product term in Eq. 12 can be negative. Rather, it describes the expected residual-stream accumulation pattern observed in pre-norm decoders: normalization stabilizes each branch input, while the unnormalized residual stream can grow with depth. Many pre-norm decoders also apply a final RMSNorm before the output readout:

$$\bar{\mathbf{h}}_t = \mathrm{RMSNorm}_{\mathrm{final}}(\mathbf{h}_t^{(L)}), \qquad \mathrm{logits}_t = \mathbf{W}_{\mathrm{vocab}}\bar{\mathbf{h}}_t. \tag{15}$$

Our latent head uses this same backbone-normalized state:

$$\mathbf{c}_t = \mathcal{F}_\theta(\bar{\mathbf{h}}_t), \qquad \hat{\mathbf{v}}_t = \mathbf{P}_k\mathbf{c}_t + \boldsymbol{\mu}. \tag{16}$$

This distinction is important. The final RMSNorm stabilizes the decoder readout scale, but it does not make $\bar{\mathbf{h}}_t$ an input-compatible vision embedding. PCA reconstruction supplies the missing input-compatibility step: it maps the normalized decoder state into coefficients of the empirical vision-embedding subspace before the latent is fed back as the next input token.

F PCA Component-Count and Capacity-Control Ablation

Table 8 varies the PCA retained-variance threshold while keeping the backbone, curated training data, 36 latent tokens, difficulty-aware supervision, and evaluation benchmarks fixed. The baseline Qwen2.5-VL-7B row has no latent head, while LH+DA uses the full latent head in the original $d = 3584$ vision-embedding space without PCA compression. All PCA rows use the same difficulty-aware policy and differ only in the number of retained PCA components induced by the retained-variance threshold.

Table 8: PCA component-count ablation under difficulty-aware latent supervision. Component ratio is measured relative to the original vision-embedding dimension $d = 3584$; the full-head LH+DA baseline predicts in this original space without PCA compression. RelMSE is the relative reconstruction error in Eq. 7. Avg-3 is the unweighted average of HRBench4K Overall, MMStar, and MathVista. Δ is measured relative to Qwen2.5-VL-7B Avg-3. Best results are in **green** and second-best in yellow.

Run	Var.	Ratio	# Comp.	RelMSE↓	Tok.	HR4K	MMStar	MVista	Avg-3	Δ
Qwen2.5-VL	–	–	–	–	0	68.13	60.46	68.50	65.70	0.00
LH+DA	1.00	100.0%	3584	–	36	71.88	61.80	68.90	67.53	+1.83
PCA-0.85+DA	0.85	7.4%	265	0.1494	36	71.00	62.60	69.30	67.63	+1.93
PCA-0.90+DA	0.90	10.6%	381	0.0999	36	71.38	61.60	70.10	67.69	+1.99
GAP (LH+PCA+DA)	0.95	17.6%	629	**0.0500**	36	**73.25**	**63.40**	**71.00**	**69.22**	**+3.52**

The uncompressed LH+DA head already improves over the base model, showing that the auxiliary latent objective is useful. However, all PCA+DA variants further improve Avg-3 relative to the uncompressed head, indicating that constraining the generated latent to an empirical vision-embedding coordinate system is helpful beyond adding a full latent head. The relative reconstruction error decreases from 0.1494 at PCA-0.85+DA to 0.0999 at PCA-0.90+DA and 0.0500 at PCA-0.95+DA, matching the expected residual variance left outside the retained PCA subspace. The downstream trend follows the same broad direction: Avg-3 rises from 67.63 and 67.69 with fewer components to 69.22 at PCA-0.95+DA. This does not imply that reconstruction error alone determines benchmark performance, but it supports the interpretation that retaining too few components discards task-relevant visual detail. At the same time, the PCA rows outperform the uncompressed LH+DA head, suggesting that the benefit is not merely from increasing latent-head capacity; the input-space PCA parameterization itself is useful. We therefore interpret PCA as a subspace-and-capacity design choice rather than as a standalone normalization operation.

G OOD Hallucination Evaluation

We additionally evaluate POPE as an OOD hallucination-sensitive sanity check. POPE uses yes/no object-recognition queries under popular, adversarial, and random negative sampling, which differs from our main benchmarks: the desired behavior is conservative binary recognition rather than producing richer visual evidence for long-form reasoning.

Table 9: OOD hallucination evaluation on POPE. Scores are reported for the overall benchmark and its popular, adversarial, and random splits. Best results are in **green** and second-best in yellow.

Method	Overall↑	Popular↑	Adversarial↑	Random↑
Qwen2.5-VL-7B	86.23	**86.34**	**85.33**	87.03
Ours (LH+DA, no PCA)	85.72	85.70	84.36	87.09
Ours-GAP (LH+PCA+DA)	**86.32**	85.76	84.49	**88.70**

Table 9 shows that the unconstrained LH+DA variant slightly underperforms the base model on overall POPE, despite improving the main perception aggregate. In contrast, GAP recovers this drop and slightly improves the overall score. We interpret this pattern as suggestive evidence that an unconstrained full latent head can over-specialize to the curated latent-supervision distribution, while the PCA reconstruction path acts as a capacity and subspace constraint by forcing generated latents to lie in the empirical vision-embedding subspace. This should not be read as a claim that GAP solves hallucination: the overall gain is mainly driven by the random split, while the popular and adversarial POPE splits remain below the base model, indicating that hallucination-sensitive OOD cases remain challenging.

H Implementation Details

Training hyperparameters. We fine-tune with AdamW, weight decay 0.01, cosine learning-rate decay, warmup ratio 0.03, peak base learning rate 1×10^{-5}, peak latent-head learning rate 1×10^{-5}, global batch size 128, per-device batch size 1, and 2 epochs. The main difficulty-aware run uses 36 latent tokens; Section J reports a latent-token-budget ablation because this choice affects both alignment capacity and inference cost. Training uses bf16, FlashAttention-2, gradient checkpointing, 16 dataloader workers, and seed 12345.

PCA computation. After preprocessing the curated 49K corpus, we split the remaining 36,985 examples into training and validation partitions using a 90/10 protocol and fit PCA exclusively on the training partition, which contains 33,279 samples. No validation examples or downstream benchmark images or embeddings are used to fit PCA. We encode each auxiliary-image occurrence with the frozen vision encoder and adaptively pool its outputs into fixed token-level embeddings, yielding approximately 992,664 embedding vectors. We center these vectors, accumulate their covariance matrix, and obtain the principal directions by eigendecomposition. The minimum rank retaining 95% of the fitting-data variance is $k = 629$. The empirical mean and PCA basis are stored and frozen during student training; the same fit is reused across checkpoints, random seeds, and downstream tasks sharing the same vision encoder and fitting data. The basis is encoder-specific. We do not refit PCA separately for individual examples, checkpoints, or evaluation benchmarks.

Latent head implementation. The latent projector is initialized from $\mathbf{P}_k^\top$ so that the head starts from the empirical vision subspace rather than an unconstrained random basis. We then refine this projection with a compact SwiGLU adapter. At inference time, ordinary text positions use the LM head, whereas latent positions use the latent head to produce PCA coefficients that are decoded by the fixed basis before feedback, as illustrated in Appendix Figure 3.

Scheduled sampling. During teacher-forced latent training, `<|latent_pad|>` positions receive the auxiliary-image vision features. When scheduled sampling is enabled as an ablation, these features can be replaced by the model's PCA-decoded predictions before a second forward pass, making training closer to inference where generated latents are fed back autoregressively. Unless otherwise stated, all results in the main tables use teacher forcing and do not use scheduled sampling.

Difficulty-aware data loading. For examples with an empirical `accuracy` field greater than τ, the loader strips the latent span and auxiliary-image placeholder targets, so the example contributes only language-modeling loss. For examples with `accuracy` at or below τ, the loader retains the latent span and auxiliary-image target embeddings for joint LM and latent-head supervision. The main experiments use $\tau = 0$, matching Eq. 4.

EMA norm calibration. The EMA norm calibration is used only as a training-free diagnostic for the norm-mismatch analysis. At inference, the running average $\bar{n}_{\text{EMA}}$ is initialized from the query

image's vision embedding norm and updated token by token before rescaling generated latents to the input vision-embedding norm regime.

I Training and Inference Cost

Training cost. The main GAP (LH+PCA+DA) model is trained on 32 GPUs for 3 hours and 55 minutes of wall-clock time. This reports student fine-tuning cost only; offline auxiliary-image generation, teacher response generation, PCA computation, and benchmark evaluation are excluded.

Offline data-construction cost. For the 49K corpus, auxiliary-image generation with Nano Banana 2 is estimated to cost approximately US$3.31K at the standard API prices used for this estimate. Teacher responses are synthesized offline with Qwen3-VL-235B-A22B-Thinking. Measured throughput on eight GPUs gives an estimated cost of approximately 274 GPU-hours for synthesizing 49K responses. These figures estimate the two data-generation stages rather than a measured end-to-end construction cost.

Latency measurement. We report wall-clock latency in seconds per sample measured inside the VLMEvalKit inference wrapper. All rows use the direct benchmark prediction files and ignore duplicate judge/output files. Because Dense Caption SFT uses the same 49K curated training data and decoding setup but removes latent tokens and auxiliary-image alignment, we use it as the primary latency baseline for relative cost. Qwen2.5-VL-7B is kept as a performance reference in the main tables, but not as the relative-latency denominator because its earlier timing was collected under an unmatched setup.

Table 10: Inference cost on HRBench4K and MathVista. Latency is wall-clock seconds per sample. Mean latency is the primary cost metric; relative latency uses Avg-2 mean latency and Dense Caption SFT as the denominator. Avg-2 performance averages HRBench4K Overall and MathVista. For latency and relative cost, lower is better; for Avg-2, higher is better. Best results are in **green** and second-best in yellow.

Method	Tok.	Mean HR/MV/Avg-2 (s)	Rel.	Avg-2
Dense Caption SFT	0	**18.81 / 28.74 / 23.78**	**1.00**×	68.54
Ours-GAP (LH+PCA+DA)	4	27.05 / 44.08 / 35.56	1.50×	70.64
Ours-GAP (LH+PCA+DA)	16	29.55 / 46.50 / 38.02	1.60×	**72.27**
Ours-GAP (LH+PCA+DA)	36	29.59 / 47.13 / 38.36	1.61×	72.13
Ours-GAP (LH+PCA+DA)	64	28.47 / 49.14 / 38.80	1.63×	71.98
Ours-GAP (LH+PCA+DA)	144	28.09 / 46.96 / 37.53	1.58×	70.83

The cost increase is moderate but non-negligible: the main 36-token setting is $1.61\times$ slower than Dense Caption SFT on Avg-2 mean latency, while improving Avg-2 performance from 68.54 to 72.13. The latency is not strictly monotonic in the latent-token budget because wall-clock time is also affected by generated response length and benchmark-specific long tails. For example, Dense Caption SFT has Avg-2 P50/P90 latencies of 10.63/30.27 seconds, while the 36-token setting has 16.10/106.71 seconds; the long tail is dominated by MathVista responses. Thus, latent tokens provide a clear accuracy-cost tradeoff rather than a free improvement, which motivates the budgeted-latent analysis in Figure 4.

J Latent-Token Budget Sweep

The number of latent tokens is a core design choice: too few tokens may under-represent auxiliary visual evidence, while too many tokens can introduce unnecessary computation and redundant self-generated visual features. We evaluate the GAP (LH+PCA+DA) model with different latent-token budgets on HRBench4K, MMStar, and MathVista, using the zero-shot Qwen2.5-VL 7B model as a reference baseline. Because latent features are arranged as a square grid with equal height and width, all positive token budgets are perfect squares.

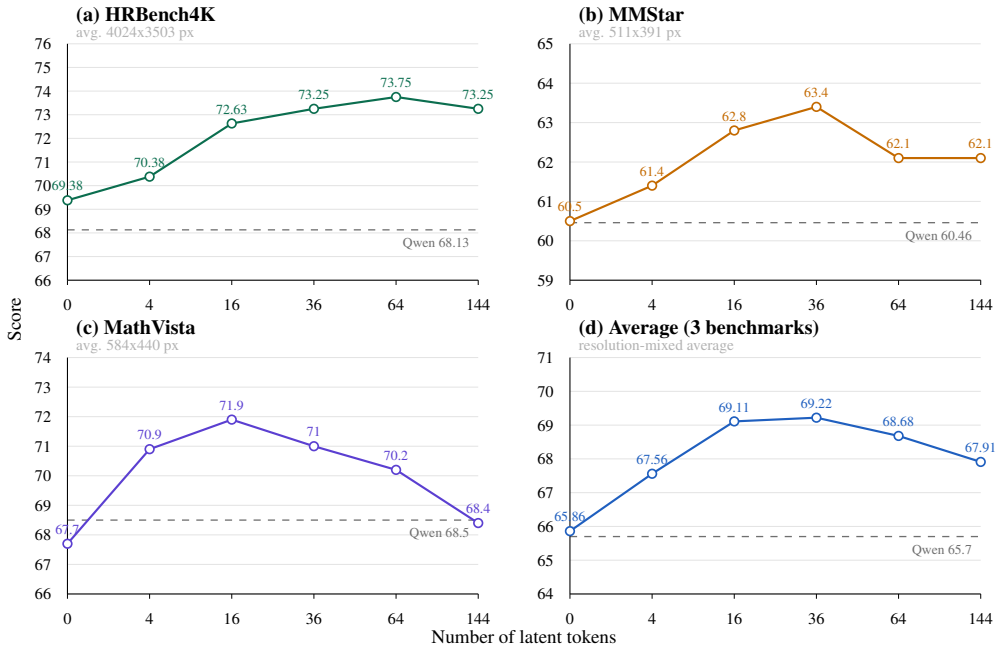


Figure 4: Effect of latent-token budget. We sweep the number of generated visual latent tokens while keeping the backbone, training data, PCA-aligned latent head, and difficulty-aware rule fixed. The gray dashed line is the Qwen2.5-VL 7B baseline. Panel subtitles report average image resolution estimated from the evaluated benchmark images. Positive token counts are square latent grids: 4=2×2, 16=4×4, 36=6×6, 64=8×8, and 144=12×12.

Table 11 reports three-evaluator-seed mean ± standard deviation results for the token-budget sweep on HRBench4K, MMStar, and MathVista. These runs cover the 36-token setting used in the main tables and nearby capacity settings. The zero-token row in this sweep is reported separately from the fixed-checkpoint feedback-suppression intervention in Table 6; the two rows should not be treated as interchangeable controls. Appendix Table 15 extends the capacity analysis to MME-RW-Lite.

Table 11: Three-evaluator-seed mean ± standard deviation results for the latent-token budget sweep in Figure 4. Per-benchmark columns report mean ± standard deviation, while Avg-3 reports the mean average over HRBench4K Overall, MMStar, and MathVista. Standard deviations reflect evaluator variability, not training-run variability. Best results are in **green** and second-best in yellow.

Latent tokens	HRBench4K	MMStar	MathVista	Avg-3
0	69.38±0.211	60.50±0.431	67.70±0.183	65.86
16	72.63±0.176	62.80±0.387	**71.90±0.215**	69.11
36	73.25±0.220	**63.40±0.340**	71.00±0.120	**69.22**
64	**73.75±0.295**	62.10±0.289	70.20±0.141	68.68
144	73.25±0.134	62.10±0.412	68.40±0.167	67.91

Figure 4 shows that latent-token capacity helps, but the effect is not monotonic. The best average over HRBench4K, MMStar, and MathVista is achieved at 36 tokens, with 16 tokens close behind, after which performance saturates or declines. HRBench4K benefits from larger latent grids: its score rises from 69.38 with zero latent tokens to 73.75 with 64 tokens, and remains high at 144 tokens. The lower-resolution benchmarks saturate earlier: MathVista peaks at 16 latent tokens, and MMStar peaks around 36 tokens before dropping at larger budgets. These results suggest that visual latent tokens should be budgeted rather than maximized.

K Data Construction Details

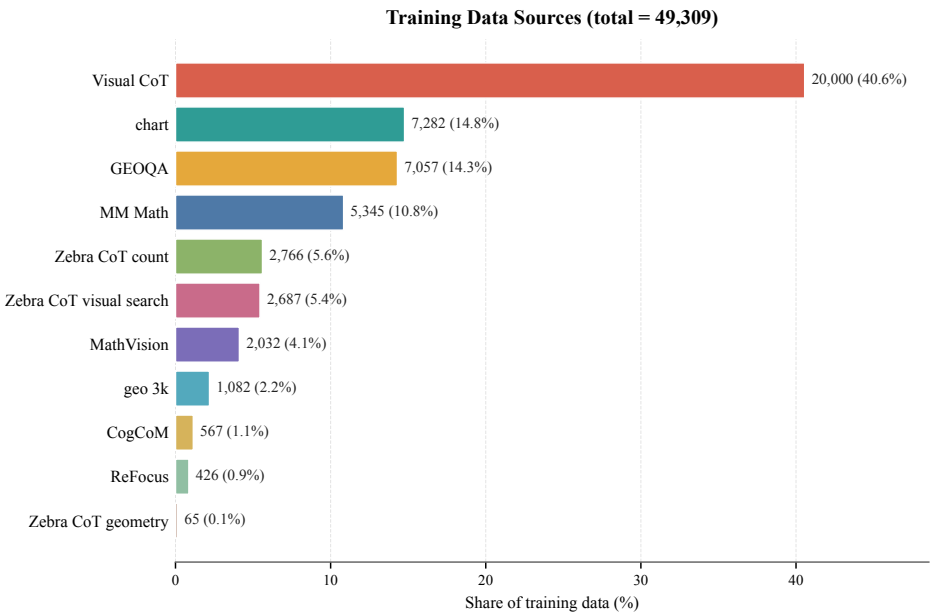


Figure 5: Composition of the 49K curated corpus before preprocessing (49,309 examples). Visual CoT is the largest source, followed by chart, GEOQA, and multimodal math data; the remaining sources emphasize geometry, counting, visual search, and focused visual reasoning.

Auxiliary image generation and response synthesis. For each training query, we first use Nano Banana 2 to generate an intermediate auxiliary image intended to expose the missing visual evidence needed by the question. We then provide the query image, question, and generated auxiliary image to Qwen3-VL-235B-A22B-Thinking, which produces the structured response containing the reasoning trace, `<latent>` span, `<parser>` description, and final answer. Only the student model is trained: the auxiliary image is converted into frozen-ViT features for latent-head supervision, while Qwen3-VL-235B-A22B-Thinking serves only as an offline teacher for response synthesis.

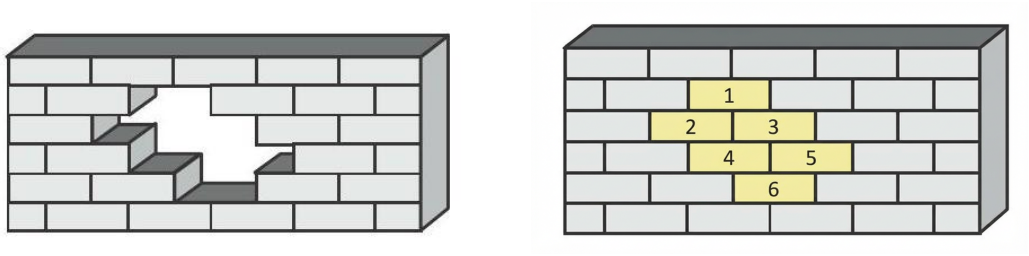
Latent serialization. The student input contains only the query image and question. The target response follows the structure `<think>` context `<latent>` auxiliary-image embedding targets `</latent>` `<parser>` dense caption `</parser>` continued reasoning `</think>` `<answer>` answer `</answer>`. In the implementation, each latent span is serialized as `<|latent_start|>` followed by fixed `<|latent_pad|>` tokens and `<|latent_end|>`. The main setting uses 36 latent tokens unless stated otherwise.

Quality filtering and supervision assignment. Quality filtering removes examples with degenerate auxiliary-image embeddings or generated `<parser>` outputs that fail a length or coherence check. Difficulty-aware assignment is a separate step: examples whose repeated-sampling accuracy exceeds τ are retained with text-only supervision, rather than removed from the corpus. For these examples, the loader strips the latent span and auxiliary-image targets, as described in Appendix H. The curated corpus contains 49,309 examples. After preprocessing, 36,985 examples remain; we then split them 90/10 into 33,279 training and 3,706 validation examples, which form the released dataset.

Leakage audit. We audit the curated training set against the evaluation benchmarks used in this paper. The audit finds no exact image duplicates and no exact normalized question-text duplicates between training and evaluation examples. We also manually inspect high-similarity cases surfaced by permissive near-duplicate heuristics. These cases primarily arise from public geometry sources such as GEOQA and Geo3K, which are also reused or adapted by existing evaluation benchmarks. We therefore avoid claiming that the training and evaluation distributions are fully disjoint, but verify that the training set does not contain exact copies of the evaluated examples.

K.1 Auxiliary-Image Supervision Example

Figure 6 shows one training example from our data construction pipeline. The original query asks: *"How many bricks are missing in the wall?"* The auxiliary image reconstructs the expected running-bond pattern and labels the six missing brick positions. This example illustrates that the teacher response provides not only a final answer, but also a human-readable description of the intended auxiliary visual evidence used to supervise the latent channel.



(a) Query image (b) Generated auxiliary image

Figure 6: Example of auxiliary-image supervision. The auxiliary image makes the missing visual evidence explicit by reconstructing the wall pattern and numbering the six missing bricks.

Qwen3-VL-235B-A22B-Thinking: teacher response excerpt

To determine how many bricks are missing, I need to reconstruct the complete wall pattern and count the gaps. The wall uses a standard running bond pattern where each brick is half-offset from the row above and below. First, I should create an auxiliary image that overlays the missing sections with the expected brick positions to clearly identify each missing brick.

<|vision_start|><|image_pad|><|vision_end|>

<parser>

- Draw the original brick wall with the damaged section.
- In the damaged area, add dashed outlines of the bricks that should be present to complete the running bond pattern.
- Label each missing brick location with a number, 1 through 6, corresponding to the auxiliary image.
- Highlight the damaged area with a light red background to distinguish it from the intact wall.
- Ensure the dashed outlines match the size and orientation of standard bricks in the wall.

</parser>

Using the pattern established in the auxiliary image, I can now map the damaged section to see how many bricks should be there. The damaged area spans multiple rows:

- In the row containing brick 1, there is one missing brick.
- In the row containing bricks 2 and 3, there are two missing bricks.
- In the row containing bricks 4 and 5, there are two missing bricks.
- In the row containing brick 6, there is one missing brick.

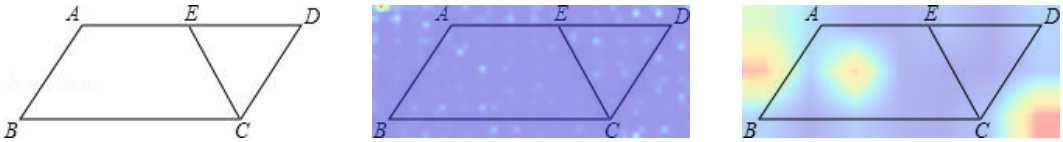
Adding these up gives $1 + 2 + 2 + 1 = 6$ missing bricks. The auxiliary image shows this by labeling exactly six missing brick positions.

<answer> The wall follows a running bond pattern, where adjacent rows are offset by half a brick. By comparing the damaged region to the intact rows and the auxiliary image, the missing bricks are labeled 1 through 6. Therefore, the total number of missing bricks is $\boxed{6}$. </answer>

This example shows how the teacher turns an image-level counting problem into a concrete visual reconstruction target. The auxiliary image supplies frozen-ViT features for latent supervision, while the <parser> text records the intended visual content of that latent target. During student training, the auxiliary image is not supplied as an image input to the student. Its frozen-ViT features fill the teacher-forced <|latent_pad|> positions and also serve as supervision targets for the latent head.

L Qualitative Inference Example

Figure 7 shows a MathVista geometry example where Qwen2.5-VL-7B predicts an incorrect answer, while our model recovers the correct auxiliary construction and answer. The question asks: *"As shown in the figure, in the parallelogram $ABCD$, CE bisects $\angle BCD$ and it intersects the AD edge at point E, and $DE = 3.0$, then the length of AB is ()."* The answer choices are: (A) 1, (B) 2, (C) 3, and (D) 6.



(a) Question image (b) Qwen2.5-VL-7B attention (c) Ours attention

Figure 7: A qualitative MathVista example. Attention maps are shown only for qualitative inspection.

Baseline failure. Qwen2.5-VL-7B retrieves a plausible theorem but grounds it to the wrong geometric configuration. The following response excerpt is edited only for formatting and math typesetting, while preserving the model's reasoning chain:

Qwen2.5-VL-7B: incorrect response excerpt

To solve this problem, we need to use the properties of a parallelogram and the angle bisector theorem.

1. **Identify the properties of the parallelogram.** In a parallelogram, opposite sides are equal in length, opposite angles are equal, and consecutive angles are supplementary.
2. **Use the angle bisector theorem.** The angle bisector theorem states that if a point divides one side of a triangle into segments proportional to the other two sides, then it is the angle bisector of the opposite angle.
3. **Apply the angle bisector theorem to triangle BCD.** Since CE bisects $\angle BCD$, it divides $\angle BCD$ into two equal angles. By the angle bisector theorem, the ratio of the segments created by the angle bisector is equal to the ratio of the other two sides.
4. **Set up the proportion.** Let $AB = x$ and $CD = x$, since opposite sides of a parallelogram are equal. According to the angle bisector theorem,
$$\frac{DE}{AE} = \frac{CD}{AB}.$$
5. **Substitute the known values.** We have $DE = 3$, $AE = AD - DE = x - 3$, $CD = x$, and $AB = x$. Therefore,
$$\frac{3}{x-3} = \frac{x}{x}.$$
6. **Solve the proportion.**
$$\frac{3}{x-3} = 1, \qquad 3 = x - 3, \qquad x = 6.$$

Therefore, the length of AB is 6. The correct answer is $\boxed{D}$.

The response applies an angle-bisector-theorem proportion, $DE/AE = CD/AB$, and then substitutes $AE = AD - DE = x - 3$ to predict $AB = 6$. It therefore selects choice D. This is geometrically invalid: E lies on AD, not on the opposite side of $\triangle BCD$, so the stated angle-bisector proportion does not follow from the diagram. This failure is representative of textually fluent but visually misgrounded reasoning: the model selects a familiar theorem, but attaches it to the wrong line segment.

Our response. Our model instead identifies the equal-angle relation induced by the parallelogram structure and the angle bisector. The following response excerpt keeps the main reasoning, latent span, parser content, and final answer:

Ours: correct response excerpt

We are given a parallelogram $ABCD$ with CE bisecting $\angle BCD$ and intersecting AD at E. We know $DE = 3.0$ and need to find AB. In a parallelogram, opposite sides are equal, so $AB = CD$. The key is to relate DE to CD using the angle bisector and the parallel lines.
First, recall that in a parallelogram, $AD \parallel BC$. Therefore, $\angle DEC = \angle ECB$ by alternate interior angles for parallel lines cut by the transversal CE. Since CE bisects $\angle BCD$, $\angle BCE = \angle DCE$. Combining these angle equalities gives $\angle DEC = \angle DCE$. Thus, $\triangle DEC$ is isosceles, and $DE = DC$. The problem states $DE = 3.0$, so $DC = 3.0$. Since opposite sides of a parallelogram are equal, $AB = CD = 3.0$.
To make this reasoning explicit, we should construct an auxiliary image that highlights the key angle equalities and the isosceles triangle.

<latent> visual latent tokens </latent>

<parser>

- Draw parallelogram $ABCD$ with $AB \parallel CD$ and $AD \parallel BC$.
- Mark point E on AD such that $DE = 3.0$.
- Draw segment CE.
- Label $\angle BCE$ and $\angle DCE$ as equal.
- Label $\angle DEC$ and $\angle DCE$ as equal to show the isosceles triangle.
- Highlight $\triangle DEC$ with a light fill.
- Mark $DE = 3.0$ on the segment.
- Indicate with equal-length ticks that $DE = DC$.
- Write $AB = CD$ near the top side.

</parser>

From the auxiliary image, we see that $\angle DEC = \angle DCE$, so $\triangle DEC$ is isosceles with $DE = DC$. Given $DE = 3.0$, it follows that $DC = 3.0$. Since $ABCD$ is a parallelogram, $AB = CD$. Therefore, $AB = 3.0$.

<answer> In parallelogram $ABCD$, $AD \parallel BC$, so $\angle DEC = \angle ECB$. Since CE bisects $\angle BCD$, $\angle ECB = \angle DCE$. Therefore, $\angle DEC = \angle DCE$, making $\triangle DEC$ isosceles with $DE = DC$. Given $DE = 3.0$, we have $DC = 3.0$. In a parallelogram, opposite sides are equal, so $AB = CD$. Thus, the length of AB is $\boxed{3.0}$, corresponding to choice C. </answer>

Interpretation. The example illustrates the intended role of visual latent reasoning. The base model follows a language-plausible but geometrically invalid theorem application, whereas our model produces a latent/parser trace that corresponds to a task-relevant auxiliary construction: highlight $\triangle DEC$, expose $\angle DEC = \angle DCE$, and infer $DE = DC$. The attention maps provide a qualitative visualization of the two runs, but we do not treat attention as causal evidence. Instead, this example complements the intervention study in Table 6, where disabling or replacing the latent rollout reduces performance relative to clean generated latents.

M Supplemental Controlled Analyses

M.1 Controlled Attribution Summary

Table 12 collects the matched comparisons that separate the contribution of the curated data from that of the training method. Each row changes one factor while keeping the backbone fixed. The curated data and the training method both contribute; among the method components, PCA alignment gives the largest gains in both aggregates (up to +1.66 Avg-P and +3.92 Avg-R), and DA adds further gains once PCA is in place.

Table 12: Controlled attribution of GAP's gains. Each comparison changes one factor while keeping the Qwen2.5-VL-7B backbone fixed; differences are computed from Tables 2 and 3.

Factor examined	**Controlled comparison**	**ΔAvg-P**	**ΔAvg-R**
Curated 49K data	LVR $\rightarrow$ LVR-49K	+0.10	+2.02
Method, data fixed	LVR-49K $\rightarrow$ GAP	+0.61	+4.28
Latent supervision	Dense Cap $\rightarrow$ LH	+0.22	+1.91
PCA alignment	LH $\rightarrow$ LH+PCA	+1.01	+3.33
PCA alignment, DA fixed	LH+DA $\rightarrow$ GAP	+1.66	+3.92
Difficulty-aware assignment	LH+PCA $\rightarrow$ GAP	+1.04	+1.49

M.2 Difficulty-Aware Assignment under Evaluator Variability

To isolate difficulty-aware assignment (DA) within the PCA-aligned recipe, we compare LH+PCA with LH+PCA+DA while holding the remaining training configuration fixed. Predictions are regenerated from the same checkpoints using a different GPU configuration from that used for the main tables, and each model's fixed predictions are scored with five evaluator seeds. Table 13 reports the resulting mean and standard deviation. GAP achieves a higher mean on all seven metrics in this comparison. These standard deviations reflect evaluator randomness on fixed outputs, not variability across training runs.

Table 13: Difficulty-aware assignment with fixed prediction sets, evaluated using five evaluator seeds. Values are mean $\pm$ standard deviation in percentage points. These statistics describe evaluator variability, not training-run variability.

Metric	LH+PCA	GAP (LH+PCA+DA)
HRBench4K Overall	72.25 ± 0.1581	73.53 ± 0.1559
HRBench4K FSP	87.75 ± 0.1574	87.80 ± 0.1418
HRBench4K FCP	56.75 ± 0.1688	59.25 ± 0.1620
MMStar	62.21 ± 0.1730	63.16 ± 0.1896
MathVista	70.06 ± 0.2348	71.20 ± 0.1643
WeMath-S	35.03 ± 0.1987	36.67 ± 0.1348
WeMath-L	52.12 ± 0.2080	56.49 ± 0.1141

M.3 PCA Fitting Distribution and Variance Retention

We examine PCA sensitivity using two equally sized, training-only fitting subsets: Global-7334 contains diverse examples, whereas Chart-7334 contains chart examples. Each subset contains 7,334 samples. Student training data and the remaining configuration are held fixed within this study; we vary the PCA fitting subset and retained-variance threshold. Table 14 reports the minimum rank selected by each threshold and the downstream scores. No validation or downstream benchmark images or embeddings are used to fit these bases.

Both fitting subsets obtain their highest Avg-2 at 95%, although the selected rank differs substantially (420 for Global-7334 vs. 216 for Chart-7334). The margin over nearby thresholds also depends on the fitting distribution: Global-90 is 0.64 points below Global-95 in Avg-2, whereas Chart-90 is only 0.05 points below Chart-95. At equal fitting-set size, Global-95 exceeds the best Chart setting on all reported metrics, so the fitting distribution affects downstream accuracy even at a fixed threshold.

Table 14: PCA sensitivity to fitting-data distribution and retained variance. Global-7334 and Chart-7334 each contain 7,334 training samples. Scores are percentages; Avg-2 averages HRBench4K Overall and MathVista. These subset-based ranks differ from the default rank of 629 fitted on the full preprocessed training partition.

Fitting set	Variance	Rank	HR Ovr.	FSP	FCP	MathVista	Avg-2
Global-7334	85%	163	67.38	83.30	51.45	65.48	66.43
Chart-7334	85%	71	68.94	83.75	54.13	65.75	67.35
Global-7334	90%	246	69.30	84.20	54.40	66.88	68.09
Chart-7334	90%	114	69.50	84.25	54.75	65.60	67.55
Global-7334	95%	420	70.92	86.25	55.58	66.53	68.73
Chart-7334	95%	216	69.50	84.50	54.50	65.70	67.60
Global-7334	99%	1104	69.63	84.13	55.13	66.60	68.12
Chart-7334	99%	618	69.75	84.25	55.25	65.00	67.38

Our main model uses one frozen basis and a shared 95% rule across benchmarks; this study does not entail benchmark-specific PCA fitting. The numerical basis remains specific to its vision encoder and fitting distribution, and should be re-estimated when those change. In particular, ranks 420 and 216 at 95% in this study should not be confused with rank 629 from the full preprocessed training partition.

M.4 Latent-Token Capacity on MME-RealWorld-Lite

Table 15 examines whether increasing the latent-token budget improves the MME-RealWorld-Lite results. Increasing the budget from 36 to 144 tokens raises Overall, Reasoning, and Perception by 2.39, 2.13, and 2.56 points, respectively, without MME-specific training data or PCA refitting. The 144-token model exceeds LVR on Overall and Reasoning and remains 0.20 points below LVR on Perception. These results show a capacity trade-off within the evaluated setting, rather than a uniform ranking across all budgets and metrics. We retain 36 tokens as the shared default used to balance perception and reasoning across benchmarks; the 144-token result is a capacity analysis, not a replacement selected separately for each benchmark. Both settings use a fixed token budget and do not implement dynamic per-example allocation.

Table 15: Latent-token capacity on MME-RealWorld-Lite. Scores are percentages. The default GAP configuration uses 36 tokens; the 144-token configuration is an additional capacity control.

Method	Overall	Reasoning	Perception
LVR	49.45	43.07	53.55
GAP (36 tokens)	47.77	43.07	50.79
GAP (144 tokens)	50.16	45.20	53.35

M.5 Geometry Subset and Information-Preservation Scope

Low-variance visual directions can contain task-relevant information, so variance retention alone does not guarantee preservation of such information. As an additional empirical check, we evaluate the 208 geometry-problem-solving examples in MathVista using five evaluator seeds. Table 16 shows a 5.39-point improvement in mean accuracy over the base Qwen2.5-VL-7B model. The reported standard deviations reflect evaluator-seed variability, not independent training runs.

Table 16: MathVista geometry-problem-solving subset (208 examples), evaluated using five evaluator seeds. Accuracy is reported as mean $\pm$ sample standard deviation (ddof=1) in percentage points.

Model	Geometry accuracy
Qwen2.5-VL-7B	70.19 ± 0.5888
GAP	75.58 ± 0.6268

GAP thus improves the evaluated geometry subset; this check alone does not show that PCA preserves every low-variance direction or that the gain comes from PCA alone. PCA constrains the reconstructed

latent embeddings, not all transformer hidden states, and the original-image tokens remain accessible during generation. Consequently, this evaluation measures task performance through the complete model rather than direct recovery of discarded PCA directions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the motivation, GAP components, and empirical scope of the paper. The claimed scope is further bounded by the limitations discussed in Appendix B.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The conclusion (Section 6) states the empirical scope, and Appendix B discusses backbone and benchmark scope, the heuristic difficulty threshold, PCA constraints including the encoder-specific basis and the dependence of the selected PCA rank on the fitting distribution, non-uniform gains and the latent-token budget trade-off, parser-latent faithfulness, and the auxiliary-image requirement.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results or proof-based claims. The equations define the PCA reconstruction, latent loss, and pre-norm motivation used by the method and analysis.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper describes the architecture, training objective, data construction, hyperparameters, PCA computation, difficulty-aware data loading, and evaluation protocol in Sections 4–5 and Appendices H–K.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The curated latent-supervision dataset and the training and evaluation code are available through `https://github.com/Qwen-Applications/GAP`, with instructions for reproducing the main experiments. The manuscript documents data construction and implementation details in Appendices K and H.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 and Appendix H specify the model, optimizer, learning rates, batch size, epochs, seed, PCA rank, latent-token setting, and inference protocol used for the experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Figure 2(a) reports shaded one-standard-deviation bands for the diagnostic norm statistics. Appendix Table 11 reports three-evaluator-seed mean $\pm$ standard deviation results for the latent-token sweep on HRBench4K, MMStar, and MathVista, including the 36-token setting used in the main tables, and Appendix Tables 13 and 16 report five-evaluator-seed mean $\pm$ standard deviation. All of these standard deviations quantify evaluator variability rather than variability across independently trained models. Remaining benchmark entries are point estimates due to compute constraints.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix I reports the compute used for the main student fine-tuning run, including GPU count and wall-clock training time, estimates the offline data-construction cost (image-generation API fees and teacher-synthesis GPU-hours), and describes the inference-latency measurement protocol.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics, uses benchmark and model-generated multimodal data, discusses limitations and broader impacts, and does not involve human-subject experiments.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix C discusses potential benefits, risks from grounded-looking incorrect answers, parser-trace limitations, and caution for high-stakes deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained foundation model, image generator, or scraped dataset requiring special release safeguards. The released assets are intended to support reproducibility of the proposed visual latent reasoning study.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Appendix D lists the existing models, datasets, benchmarks, generation services, and comparison assets used in the paper, including upstream URLs and license or terms-of-use status where declared. Assets without standard license declarations or with restrictive terms are used only for evaluation or metadata/script-based release, not redistributed as benchmark content.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces a curated 49K latent-supervision dataset and documents its composition, construction, filtering, leakage audit, and examples in Appendix K. The dataset is publicly released through `https://github.com/Qwen-Applications/GAP` with a dataset card describing its sources, construction, and licenses, together with the code and its documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

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

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval or equivalent review is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LLM and MLLM usage is central to the method: Qwen2.5-VL is the student backbone and Qwen3-VL-235B-Thinking is used as an offline teacher for response synthesis, as described in Section 4.1 and Appendix K.

Guidelines:

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