From Answer-Aware to Context-Only: Bridging Prior and Posterior for Multimodal Latent Reasoning

Anonymous ACL submission

Abstract

Latent reasoning has recently emerged as a powerful alternative to explicit textual reasoning in Multimodal Large Language Models (MLLMs), leveraging continuous representations to unlock a vastly expanded exploration space for hard-to-verbalize reasoning pathways. However, existing methods often constrain this space via handcrafted supervision or predefined semantics, confining models to sub-optimal, human-prescribed pathways. To fully unleash the potential of latent exploration and enable models to autonomously discover optimal pathways using target answers without handcrafted supervision, we propose **P**rototype-based r**E**ciprocal l**A**tent **R**easoning **L**earning (PEARL). Specifically, PEARL bridges a context-only prior branch and an answer-aware posterior branch through a Shared Prototype Interface. We introduce Reciprocal Divergence Constraints, modulated by an evidence-guided confidence weight, to bidirectionally align the two branches. This structured mechanism safely anchors the posterior to realizable latent modes while enabling it to actively guide the prior, stably distilling autonomously discovered reasoning structures for inference. Extensive experiments across five benchmarks demonstrate that PEARL decisively outperforms existing discrete and latent baselines, yielding comprehensive average gains across core spatial (+5.22) and complex cognitive (+8.92) tasks.

1 Introduction

Multimodal large language models (MLLMs) (Comanici et al., 2025; Bai et al., 2025a) have achieved strong performance on many vision-language tasks, but complex reasoning remains a major bottleneck (Liu et al., 2024; Cheng et al., 2026). Many challenging multimodal problems require complex thinking beyond direct perception, demanding the model to integrate subtle visual cues and derive implicit intermediate conclusions before producing the final answer (Wang et al., 2024; Bai et al., 2025b).

A dominant line of recent work improves reasoning through explicit intermediate processes, such as chain-of-thought (CoT) prompting (Huang et al., 2026) or tool use (Zheng et al., 2026). While effective, these methods assume that intermediate reasoning can be expressed through discrete symbolic forms, such as natural language tokens or predefined tool calls. This assumption is often too restrictive for multimodal reasoning, where useful intermediate computation may be difficult to verbalize or may involve fine-grained cross-modal interactions that are not naturally expressible in language (Xu et al., 2026; Guan et al., 2024).

This limitation has motivated growing interest in latent reasoning (Hao et al., 2024; Team, 2025; Li et al., 2026b), where intermediate computation is carried out through latent tokens rather than explicit textual traces (Ray et al., 2025; Wang et al., 2025b). In principle, this continuous latent space should offer flexibility for representing implicit and hard-to-verbalize reasoning processes. However, many existing methods still rely on handcrafted supervision signals or manually assigned token semantics, effectively prescribing specific routes in latent space according to human assumptions. Such designs restrict the model's exploration of possible reasoning pathways, limit representation coverage, and ultimately prevent it from discovering latent structures that better connect multimodal input to the target answer.

We argue that latent reasoning pathways should not be manually predefined. Instead, the model should be allowed to discover its own pathway from input to answer. In other words, ***what we need to provide is not the pathway itself, but a structured learning mechanism that allows effective pathways to emerge.*** However, achieving this is nontrivial. A naive solution is to expose the

model to the target answer during training and let it infer useful latent states from both input and answer (Lin et al., 2025; Wang et al., 2026a). Yet this easily leads to *answer leakage*: the model may directly encode answer information into latent tokens, bypass the intended reasoning process, and thereby hurt generalization at inference time, when the answer is unavailable. This creates a central challenge for latent reasoning: how can we exploit answer-aware supervision during training without sacrificing inference-time feasibility?

To address this challenge, we propose **P**rototype-based r**E**ciprocal l**A**tent **R**easoning **L**earning (**PEARL**), a latent reasoning framework for MLLMs. PEARL uses two asymmetric training branches: a *context-only prior* branch that matches inference-time conditions, and an *answer-aware posterior* branch that has access to the target answer and can infer richer latent representations. Rather than allowing the two branches to form independent latent spaces, PEARL couples them through a **Shared Prototype Interface** (SPI) module, which maps branch-specific latent queries into a shared prototype space. This design enables informative posterior guidance while constraining the learned latent structure to remain realizable by the prior at inference time.

To further control answer leakage, we introduce **Reciprocal Divergence Constraints** to bidirectionally couple the prior and posterior branches. The posterior guides the prior toward more informative latent representations. At the same time, it is regularized to remain within latent modes that the prior can realize, preventing drift toward answer-specific shortcuts. This reciprocal coupling makes posterior supervision more stable and transferable, enabling the prior to learn effective latent reasoning from context alone. As a result, PEARL can exploit answer-aware supervision during training without degenerating into direct answer copying.

Our contributions are summarized as follows:

- We propose PEARL, a framework that leverages an answer-aware posterior to guide a context-only prior, coupled via a Shared Prototype Interface to ensure inference-time feasibility.
- We introduce Reciprocal Divergence Constraints to bidirectionally regularize the prior and posterior, mitigating answer leakage while enabling the emergence of powerful latent reasoning.
- PEARL decisively outperforms baselines across five benchmarks, delivering massive average gains in spatial (+5.22) and cognitive (+8.92) reasoning, strong out-of-distribution robustness, and validated through extensive ablations.

2 Related Works

2.1 Explicit Intermediate Reasoning

A dominant paradigm enhances MLLM reasoning via explicit intermediate steps, such as chain-of-thought (CoT), rationale distillation, and tool augmentation (Wei et al., 2022; Zhang et al., 2023; Hsieh et al., 2023; Hong et al., 2025; Jiang et al., 2025; Ni et al., 2025). Recent advances further optimize these symbolic trajectories using reinforcement learning for long-form reasoning (e.g., Vision-R1 (Huang et al., 2026) and PAPO (Wang et al., 2026b)), or explicitly ground them in visual evidence via "thinking with images" (Su et al., 2025; Sarch et al., 2025; Chen et al., 2025; Chung et al., 2025; Zhang et al., 2025) and interactive frameworks like PixelReasoner (Su et al., 2026) and DeepEyes (Zheng et al., 2026).

Despite their success, these methods share a fundamental limitation: they mandate discrete symbolic representations (e.g., text tokens or tool calls) for intermediate states. This discrete bottleneck restricts reasoning capacity, particularly for complex visual computations that defy natural language verbalization. In contrast, our approach bypasses this limitation by executing intermediate inference entirely within the continuous latent space.

2.2 Latent Reasoning

Latent reasoning executes intermediate computations within continuous hidden spaces, offering superior flexibility for implicit, hard-to-verbalize multimodal processes (Goyal et al., 2024; Pfau et al., 2024; Wang et al., 2025b; Butt et al., 2025; Geiping et al., 2025; Hao et al., 2024; Shen et al., 2025; Wang et al., 2025a). Early formulations like Pause Tokens (Goyal et al., 2024) allocate internal compute via silent steps, while subsequent advances (e.g., Mull-Tokens (Ray et al., 2025), LVR (Li et al., 2026b), Monet (Wang et al., 2025b), and Coconut (Hao et al., 2024)) integrate continuous latent tokens into autoregressive generation, bypassing discrete text emission. However, these methods rely on restrictive human priors, such as handcrafted roles or fixed semantics. In contrast, our prior-posterior framework allows the model to autonomously acquire its reasoning structure while guaranteeing inference-time realizability.

2.3 Prior-Posterior Latent Learning

Our approach is grounded in prior-posterior latent learning (Kingma and Welling, 2013; Sohn et al., 2015; Hu et al., 2023; Zhou et al., 2025). While recent multimodal extensions (e.g., ReGuLaR (Wang et al., 2026a) and RAVR (Lin et al., 2025)) employ posterior-guided trajectories, they align branches using explicit traces or answer-informed targets. This inadvertently allows the answer-aware posterior to exploit trivial shortcuts, leading to severe answer leakage. To resolve this, our method bidirectionally couples the prior and posterior via a Shared Prototype Interface (SPI) and Reciprocal Divergence Constraints. This prevents drift into answer-specific shortcuts, ensuring posterior supervision remains informative during training and fully realizable by the context-only prior at inference.

3 Method

3.1 Overview

We propose **P**rototype-based r**E**ciprocal l**A**tent **R**easoning **L**earning (**PEARL**), a framework for latent reasoning in MLLMs. As shown in Figure 1(a), PEARL consists of three components: (i) an answer-aware **posterior branch** and a context-only **prior branch**, (ii) a **Shared Prototype Interface (SPI)** module equipped with a learnable prototype codebook, and (iii) **Reciprocal Divergence Constraints** that bidirectionally couple the two branches during training.

PEARL is based on asymmetric information access. During training, the posterior branch conditions on both the multimodal context and the target answer to infer informative latent representations, while the prior branch only observes the context and predicts latents under the same setting used at inference time. Both branches generate latent queries that are routed through the SPI, which maps them into a shared prototype space and produces refined latent embeddings for decoding.

Training is guided by two reciprocal objectives. The **alignment objective** regularizes the posterior within latent modes supported by the prior, while the **consistency objective** encourages the prior to match the informative posterior. At inference time, only the prior branch is used.

3.2 Prior and Posterior Branches

The prior and posterior branches share the same MLLM backbone but differ in their conditioning information and branch-specific Query Heads. Given a multimodal input consisting of an image I and a textual prompt T, we first convert them into a unified context sequence $\mathbf{X}_{\text{ctx}} = [x_1, \ldots, x_n]$ following standard MLLM practices. Building upon this shared context, each branch constructs its own specific input sequence $\mathbf{X}^b$ by appending different functional tokens (e.g., target answer tokens and latent tokens), where $b \in \{p, q\}$ denotes the prior or posterior branch. This sequence is then processed by the shared backbone:

$$\mathbf{H}^b = \text{MLLM}(\mathbf{X}^b) \in \mathbb{R}^{T_b \times d}, \tag{1}$$

where $\mathbf{H}^b$ denotes the hidden states produced by the backbone, with T_b being the sequence length and d the hidden dimension. Let $h_t^b \in \mathbb{R}^d$ denote the hidden state vector at position t from $\mathbf{H}^b$. We then extract the hidden states corresponding to the k latent token positions, indexed by $L_1, \ldots, L_k$:

$$\mathbf{H}_L^b = [h_{L_1}^b, \ldots, h_{L_k}^b] \in \mathbb{R}^{k \times d}. \tag{2}$$

These hidden states are mapped to latent queries by a branch-specific Query Head:

$$\mathbf{Z}^b = f_b(\mathbf{H}_L^b) \in \mathbb{R}^{k \times d_z}, \tag{3}$$

where $f_b(\cdot)$ is implemented as an RMSNorm followed by a SwiGLU projection (Shazeer, 2020; Zhang and Sennrich, 2019). The SPI then maps the latent queries into latent embeddings:

$$\mathbf{L}'^b = \text{SPI}(\mathbf{Z}^b, \mathcal{E}) \in \mathbb{R}^{k \times d}, \tag{4}$$

where $\mathcal{E}$ denotes the shared prototype codebook.

Prior branch. The prior branch matches the inference-time setting and only observes the multimodal context together with the latent tokens:

$$\mathbf{X}^p = [x_1, \ldots, x_n,\ L_1, \ldots, L_k], \tag{5}$$

where $x_1, \ldots, x_n$ denote the context sequence elements derived from the visual and textual inputs, and $L_1, \ldots, L_k$ denote the placeholder tokens inserted for latent reasoning. Its latent embeddings $\mathbf{L}'^p$ are therefore predicted from context alone.

Posterior branch. The posterior branch is used only during training and additionally conditions on the target answer. Under causal masking, we place the answer tokens before the latent tokens so that the latent positions can attend to them:

$$\mathbf{X}^q = [x_1, \ldots, x_n,\ a_1, \ldots, a_m,\ L_1, \ldots, L_k],$$

where $a_1, \ldots, a_m$ denote target answer tokens. This ordering yields richer latent embeddings $\mathbf{L}'^q$.

3.3 Shared Prototype Interface

To prevent the prior and posterior branches from drifting into disconnected latent spaces, we introduce the **Shared Prototype Interface (SPI)** to

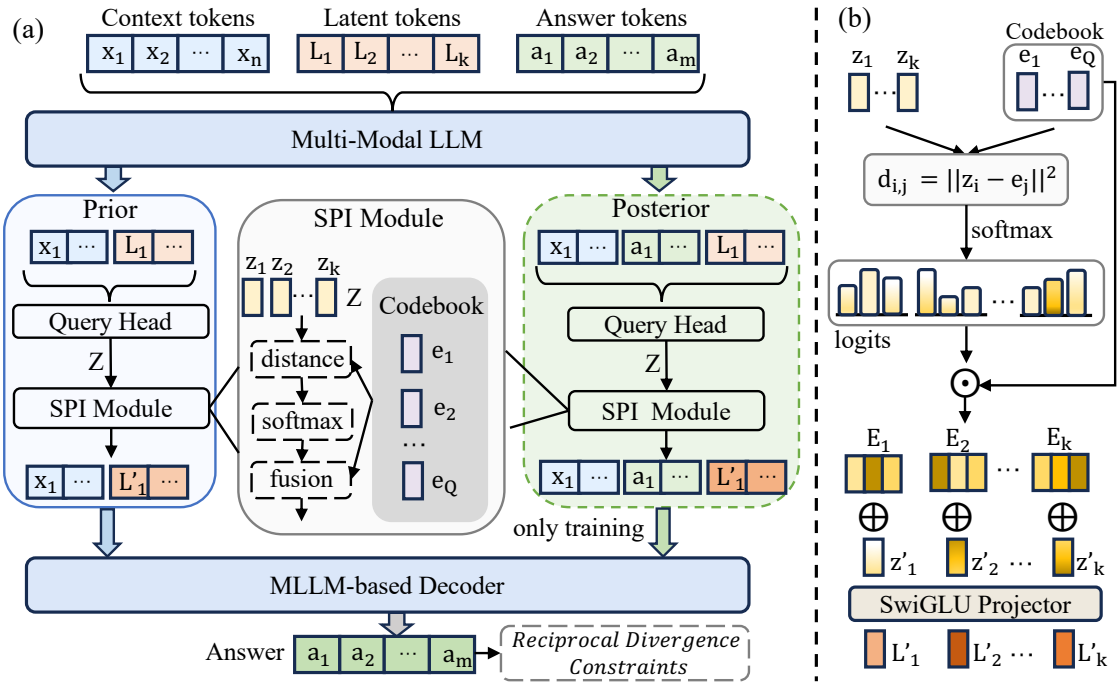


Figure 1: **PEARL framework and the SPI module. (a)** Overview of PEARL. A context-only prior branch and an answer-aware posterior branch are trained jointly through a shared prototype-based latent interface. Both branches produce latent reasoning tokens via the SPI module, while only the prior branch is used at inference time. Training is guided by reconstruction, alignment, and consistency losses. **(b)** Structure of the SPI Module. Latent queries are matched against a shared codebook using RBF-based distance scores, transformed into soft assignment weights, and fused with prototype representations to generate refined latent reasoning tokens.

serve as a common latent bottleneck between the two branches. Unlike unconstrained continuous spaces prone to arbitrary drift, or hard discrete codebooks suffering from optimization instability (Van Den Oord et al., 2017), our prototype-based approach (Takida et al., 2022; Snell et al., 2017) provides a structured yet fully differentiable latent space. Crucially, by expressing reasoning states as soft combinations of shared semantic modes, it simplifies the prior-posterior alignment: the two branches only need to synchronize their discrete distributions over a finite set of prototypes rather than matching unconstrained high-dimensional vectors. Given branch-specific latent queries:

$$\mathbf{Z}^b = [z_1^b, \ldots, z_k^b] \in \mathbb{R}^{k \times d_z}, \tag{6}$$

where $b \in \{p, q\}$ denotes the prior or posterior branch, and z_i^b is the query corresponding to the i-th latent token of branch b, the SPI maps each query z_i^b into a shared prototype space defined by a learnable codebook:

$$\mathcal{E} = \{e_1, \ldots, e_Q\}, \qquad e_j \in \mathbb{R}^{d_z}, \tag{7}$$

where Q is the codebook size and e_j denotes the j-th prototype embedding. The codebook $\mathcal{E}$ is shared by both the prior and posterior branches across all latent positions, and is jointly optimized with all other model parameters.

For each latent query z_i^b, we compute its squared Euclidean distance $d_{i,j}^b$ to the j-th prototype e_j:

$$d_{i,j}^b = \|z_i^b - e_j\|_2^2. \tag{8}$$

We directly use the negative squared Euclidean distance as the assignment logit, naturally assigning higher unnormalized scores to closer prototypes: $\ell_{i,j}^b = -d_{i,j}^b$. We group the scalar logits for the i-th query into a logit vector $\boldsymbol{\ell}_i^b = [\ell_{i,1}^b, \ldots, \ell_{i,Q}^b] \in \mathbb{R}^Q$. To obtain a differentiable assignment weight vector $\boldsymbol{\alpha}_i^b = [\alpha_{i,1}^b, \ldots, \alpha_{i,Q}^b]$ over the codebook, we apply a temperature-scaled softmax function directly to the logits:

$$\alpha_{i,j}^b = \frac{\exp(\ell_{i,j}^b/\tau)}{\sum_{t=1}^{Q} \exp(\ell_{i,t}^b/\tau)}, \tag{9}$$

where the temperature parameter τ controls the sharpness of the assignment distribution. During inference, we simply set $\tau = 1$ to recover the standard softmax operation.

The resulting assignment weights are used to compute a soft prototype for each latent position:

$$z'^b_i = \sum_{j=1}^{Q} \alpha_{i,j}^b e_j. \tag{10}$$

This aggregated prototype is projected back to the decoder hidden space through a SwiGLU projector:

$$L'^{b}_{i} = g(z'^{b}_{i}), \tag{11}$$

where $g(\cdot)$ denotes the latent-to-hidden projection module. Collecting the outputs for all latent positions gives the refined latent embeddings:

$$\mathbf{L}'^{b} = [L'^{b}_{1}, \ldots, L'^{b}_{k}] \in \mathbb{R}^{k \times d}. \tag{12}$$

After obtaining $\mathbf{L}'^{b}$, we inject these refined latent embeddings back into the sequence at the corresponding latent token positions. The resulting hidden sequence is then used by the decoder to autoregressively predict the target answer. In this way, the latent variables influence generation only through the latent token slots, preserving a clear bottleneck between latent inference and answer decoding.

By forcing both prior and posterior queries to pass through the same prototype space, the SPI prevents the two branches from forming disconnected latent spaces. Instead, both branches must express their latent reasoning states through shared prototype modes. This makes posterior guidance more structured and transferable, while keeping the learned latent space realizable by the prior at inference time.

3.4 Reciprocal Divergence Constraints

To bridge the gap between the answer-aware posterior and the context-only prior, PEARL couples the two branches through a bidirectional objective in the shared prototype space. Since the latent variables are parameterized as categorical distributions over the Shared Prototype Interface (SPI), aligning the prior distribution p (parameterized by the assignment weights $\boldsymbol{\alpha}^{p}$) and the posterior distribution q (parameterized by $\boldsymbol{\alpha}^{q}$) requires carefully designed geometric constraints. We implement this through **Reciprocal Divergence Constraints**, where the stop-gradient operator ($\mathrm{sg}(\cdot)$) is applied to the second argument of each Kullback–Leibler (KL) divergence.

Posterior alignment via prior anchoring. We first regularize the answer-aware posterior to remain close to latent modes that are already supported by the context-only prior:

$$\mathcal{L}_{\text{align}} = \mathcal{D}_{\mathrm{KL}}(q \,\|\, \mathrm{sg}(p)). \tag{13}$$

Because the stop-gradient is applied to the prior, this term updates only the posterior branch. Intuitively, the prior provides a context-conditioned structural reference, and the KL term encourages the posterior to avoid assigning substantial probability mass to prototype modes that the prior considers highly unlikely. As a result, answer-aware latent inference is softly anchored to modes that remain realizable by the prior at inference time.

Prior consistency via posterior distillation. Conversely, the prior must learn to approach the more informative latent structure inferred by the posterior:

$$\tilde{\mathcal{L}}_{\text{consist}} = \mathcal{D}_{\mathrm{KL}}(p \,\|\, \mathrm{sg}(q)). \tag{14}$$

Here the posterior is treated as a fixed teacher target within this loss term, so the gradient flows only through the prior. This encourages the prior to increase probability on prototype modes favored by the answer-aware posterior.

Directly applying this consistency objective can be unstable early in training, especially when the posterior is much sharper than the prior. To improve robustness, we modulate the distillation strength using an evidence-guided confidence weight derived from the prior's own reconstruction quality. Let $\bar{\mathcal{L}}^{p}_{\text{rec}}$ denote the average next-token prediction loss of the target answer a of length m, computed using the prior latent embeddings $\mathbf{L}'^{p}$:

$$\bar{\mathcal{L}}^{p}_{\text{rec}} = -\frac{1}{m} \sum_{t=1}^{m} \log P(a_t \mid a_{<t}, \mathbf{X}_{\text{ctx}}, \mathbf{L}'^{p}). \tag{15}$$

where $P(\cdot)$ denotes the conditional next-token probability predicted by the MLLM decoder. We then define the confidence weight as

$$w = \mathrm{sg}\big(\exp\big(-\bar{\mathcal{L}}^{p}_{\text{rec}}\big)\big)\,, \tag{16}$$

which corresponds to the geometric mean likelihood of the target answer under the prior latent state. The final consistency loss is:

$$\mathcal{L}_{\text{consist}} = w \cdot \tilde{\mathcal{L}}_{\text{consist}}. \tag{17}$$

This weighting strengthens posterior guidance on samples where the prior provides strong generative evidence, while down-weighting potentially unreliable distillation signals on harder samples.

Taken together, these two objectives form a reciprocal teacher-student coupling: the prior constrains the posterior to stay within realizable latent modes, while the posterior guides the prior toward more informative ones. This bidirectional design reduces representation drift and encourages both branches to coordinate through a shared subset of prototype modes. Further theoretical discussion of the associated optimization geometry and the role of the confidence weight w is provided in Appendix B.

Final objective. The decoder is trained with the standard next-token prediction loss on the target

Method	Venue	V*			HRBench4K			HRBench8K			Avg.
		Overall	Attribute	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	
Proprietary Models											
GPT-4o	-	67.50	72.20	60.50	59.00	70.00	48.00	55.50	62.00	49.00	60.67
Open-Source Models											
Qwen2.5-VL-7B	-	76.44	77.39	75.00	68.00	80.25	55.75	63.75	74.25	53.25	69.40
+ SFT	-	81.68	83.48	78.95	68.38	78.28	58.50	61.63	70.75	52.50	70.56
+ SFT + GRPO	-	78.53	78.26	78.95	70.00	83.25	56.75	66.75	78.00	55.50	71.76
PixelReasoner	NeurIPS'25	81.15	80.87	81.58	72.00	84.00	**60.00**	66.12	76.50	**55.75**	73.09
DeepEyes	ICLR'26	**83.25**	**84.35**	81.58	71.25	83.75	58.75	65.13	77.00	53.25	73.21
LVR	ICLR'26	81.15	80.00	**82.89**	70.62	83.50	57.75	64.12	77.25	51.00	71.96
Mull-Tokens	CVPR'26	79.06	81.58	77.49	70.25	86.50	54.00	65.75	81.00	50.50	71.69
Ours	-	82.72	83.48	81.58	**72.38**	**88.00**	56.75	**68.75**	**83.00**	54.50	**74.62**
Absolute gain	-	+6.28	+6.09	+6.58	+4.38	+7.75	+1.00	+5.00	+8.75	+1.25	+5.22

Table 1: Performance on V*, HRBench4K, and HRBench8K. "Avg." is the mean Overall score across benchmarks. Best and second-best open-source results are highlighted in **bold** and underlined. The last row shows absolute gains over Qwen2.5-VL-7B. Our method outperforms both "Discrete Reasoning" and "Latent Reasoning".

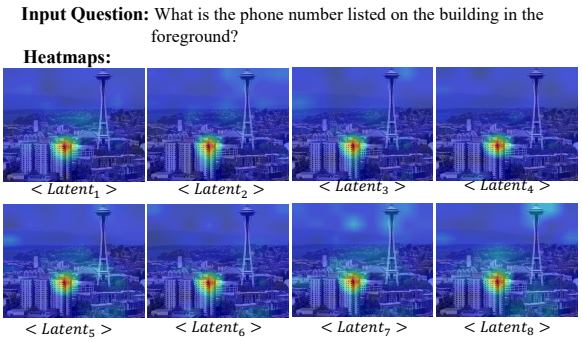


Figure 2: Sensitivity of latent reasoning tokens to patch-level image perturbations.

answer using the posterior latents:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{m}\sum_{t=1}^{m} \log P(a_t \mid a_{<t}, \mathbf{X}_{\text{ctx}}, \mathbf{L}'^{q}). \quad (18)$$

The full training objective unifies the generative reconstruction loss with latent geometric constraints:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{consist}}. \quad (19)$$

This reciprocal design couples the prior and posterior within a shared prototype space. By restricting the posterior to prior-supported modes and distilling guidance back, it balances training informativeness with inference realizability.

4 Experiments

Implementation details. We implement PEARL on top of Qwen2.5-VL-7B-Instruct (Bai et al., 2025a). Training uses a mixture of multimodal reasoning datasets, including Visual-CoT (Shao et al., 2024), ReFocus (Fu et al., 2025), CogCoM (Qi et al., 2025), and Zebra-CoT (Li et al., 2026a). We use $k = 8$ latent tokens and codebook size $Q = 512$. Following (Li et al., 2026b; Wang et al., 2025b), the vision encoder is frozen during training. Full details are provided in Appendix C.3.

Experiment Benchmarks. We structure our experiments across three axes: (1) Fine-grained perception: We use V* (Wu and Xie, 2024), HRBench4K, and HRBench8K (Wang et al., 2025c) to assess high-resolution processing and dense feature extraction. (2) Complex visual reasoning: We employ the BLINK (Fu et al., 2024) benchmark to test core visual cognition and spatial logic. (3) OOD robustness: We evaluate abstract generalization using the VisualPuzzles (Song et al., 2025) benchmark (results in Appendix D).

Baselines. For fair comparison, we benchmark against state-of-the-art MLLMs built upon the same Qwen2.5-VL-7B foundation. We categorize these into two paradigms: (1) **Discrete Reasoning** (Vision-R1 (Huang et al., 2026), PAPO (Wang et al., 2026b), PixelReasoner (Su et al., 2026), DeepEyes (Zheng et al., 2026)); and (2) **Continuous Latent Reasoning** (LVR (Li et al., 2026b), Mull-Tokens (Ray et al., 2025)). Detailed descriptions are deferred to Appendix C.2.

4.1 Main Results

As shown in Table 1, PEARL achieves the highest average score among open-source models, surpassing the Qwen2.5-VL-7B baseline by +5.22 points overall. It exhibits consistent improvements across all major benchmarks, with notable gains on HRBench4K (+4.38) and HRBench8K (+5.00). Specifically, massive gains on fine-grained FSP

Method	Venue	Counting	IQ-Test	JigSaw	Obj. Loc.	Art Style	Vis. Sim.	Rel. Depth	Avg.
Qwen2.5-VL-7B	-	68.33	20.00	45.33	49.18	64.96	82.96	72.58	57.62
PixelReasoner	NeurIPS'25	68.33	26.00	72.00	54.10	67.52	85.93	80.65	64.93
PAPO	ICLR'26	67.50	22.67	66.67	**56.56**	64.96	85.19	78.32	63.12
Vision-R1	ICLR'26	68.33	27.33	54.67	48.36	65.81	71.85	75.00	58.76
LVR	ICLR'26	65.83	26.00	52.67	50.00	**70.94**	82.22	77.42	60.73
Mull-Tokens	CVPR'26	68.33	31.33	69.33	47.54	62.42	80.74	**83.87**	63.37
Ours	-	**70.83**	**31.33**	**74.00**	55.74	68.38	**88.10**	77.42	**66.54**
Absolute gain	-	+2.50	+11.33	+28.67	+6.56	+3.42	+5.14	+4.84	+8.92

Table 2: Performance on the BLINK benchmark for complex visual reasoning. "Obj. Loc." denotes Object Localization, "Vis. Sim." denotes Visual Similarity, and "Rel. Depth" denotes Relative Depth. Avg. is the average score over all selected benchmarks. The last row shows absolute gains over Qwen2.5-VL-7B.

metrics (+7.75 and +8.75 on HRBench4K/8K-FSP) demonstrate significantly enhanced spatial perception. Furthermore, Table 2 underscores PEARL's broad cognitive capacity across diverse reasoning categories. Achieving the highest average score of 66.54 (+8.92 over the baseline), PEARL decisively outperforms all evaluated discrete and latent methods. It excels particularly in tasks demanding complex structural understanding and logical deduction, securing striking improvements in JigSaw (+28.67) and IQ-Test (+11.33), while also establishing new state-of-the-art results in Counting and Visual Similarity. Ultimately, these comprehensive gains confirm that reciprocal latent coupling systematically broadens the model's core cognitive abilities alongside its fine-grained perception.

Advantages Over Discrete Reasoning. Compared to *discrete reasoning* baselines (e.g., PixelReasoner, DeepEyes, PAPO, Vision-R1), PEARL delivers stronger, more balanced performance without explicit token-level traces. This indicates our framework captures reasoning structures while circumventing the inference overhead and brittleness typical of discrete CoT chains or tool use.

Advantages Over Latent Reasoning. PEARL also outperforms prior *latent reasoning* models (LVR, Mull-Tokens), proving the superiority of structured latent coupling over unconstrained supervision. We attribute these gains to two key designs: (1) the Shared Prototype Interface, which enables seamless prior-posterior communication via a reusable structure; and (2) Reciprocal Divergence Constraints, which effectively bridge the train-test gap between answer-aware posterior inference and context-only prior prediction.

Out-of-Distribution Generalization. Beyond in-domain evaluations, PEARL demonstrates robustness against distribution shifts. It delivers competitive performance on the OOD VisualPuzzles benchmark (Appendix D), confirming that PEARL's architectural constraints prevent overfitting to dataset biases and foster broadly generalizable abilities.

Interpretability and Visual Grounding. Finally, patch-level perturbation sensitivity analysis confirms PEARL's strong visual grounding. As shown in Figure 2, perturbing question-relevant foregrounds triggers the largest response variations in latent tokens, whereas background perturbations have minimal impact. Furthermore, distinct sensitivity patterns across different latent tokens suggest functional specialization within the latent reasoning process (further analyzed in Appendix F).

4.2 Ablation Study

We ablate key PEARL components: reciprocal training objectives, divergence constraints, Shared Prototype Interface, and latent configurations.

Effect of Latent Configuration. Table 3 ablates the latent token count and codebook size. We observe that moderate token counts (8 or 16) yield optimal performance: too few restrict expressive capacity, while excessive tokens introduce redundancy and optimization challenges. Similarly, intermediate codebook sizes (1024–4096) provide the best trade-off. Overly small codebooks limit latent diversity, whereas excessively large ones hinder prototype reuse. These findings highlight that PEARL requires a carefully balanced latent capacity across both token budget and prototype space.

Effect of Training Objectives. As observed in Table 4, the direct incorporation of $\mathcal{L}_{\text{con}}$ results in a noticeable performance degradation. This is primarily due to a unidirectional optimization where the prior is forced to align with an unconstrained posterior, inducing posterior information leakage that prevents the prior from learning robust reasoning trajectories. Although the evidence-guided confidence weight w activates the potential

Setting	Number of latent tokens					Codebook size					
	4	8	12	16	20	512	1024	2048	4096	5120	7168
V*	82.20	82.72	81.68	82.72	81.15	82.20	82.72	82.20	82.72	81.68	81.15
HRBench4K	71.88	72.38	71.62	71.88	71.00	72.38	72.50	72.25	72.38	72.12	71.75
HRBench8K	67.88	68.75	67.12	68.12	67.88	67.88	68.00	67.62	68.75	68.00	67.00

Table 3: Ablation study of latent configuration. We vary the number of latent tokens and the codebook size while keeping all other settings fixed.

$\mathcal{L}_{\text{rec}}$	$\mathcal{L}_{\text{con}}$	w	$\mathcal{L}_{\text{ali}}$	V*	HRBench4K	HRBench8K
✓	×	×	×	81.15	70.88	64.75
✓	✓	×	×	78.01	69.00	64.38
✓	✓	✓	×	76.96	71.38	66.75
✓	✓	✓	✓	82.72	72.38	68.75

Table 4: Ablation study on training objectives. We start with the base reconstruction loss $\mathcal{L}_{\text{rec}}$ and progressively evaluate the contributions of the consistency loss $\mathcal{L}_{\text{con}}$ (regulated by the evidence-guided weight w) and the alignment loss $\mathcal{L}_{\text{ali}}$.

sg_{con}	w	sg_{ali}	V*	HRBench4K	HRBench8K
×	×	×	80.63	70.88	67.50
✓	×	×	81.15	71.75	68.12
✓	✓	×	81.15	72.12	68.25
✓	✓	✓	82.72	72.38	68.75

Table 5: Ablation study on stop-gradient (sg) operations and confidence weighting (w) within the reciprocal divergence constraints.

of $\mathcal{L}_{\text{con}}$ and yields positive gains on challenging tasks (e.g., a 2% improvement on HRBench8K), the risk of leakage persists in simpler scenarios such as V*. Crucially, by incorporating $\mathcal{L}_{\text{ali}}$ to constrain the posterior toward the prior, we effectively mitigate the information leakage while preserving the posterior's valid reasoning guidance. This dual-constraint strategy leads to a holistic performance enhancement, elevating V* from 81.15 to 82.72 and HRBench8K from 64.75 to 68.75.

Effect of Stop-Gradient and Confidence Weighting. Table 5 ablates stop-gradient placement and the confidence weight w within the reciprocal divergence constraints. Unilateral stop-gradients outperform fully unconstrained updates, indicating that directional matching prevents the instability of jointly optimizing toward moving targets. Incorporating w consistently boosts performance, particularly when paired with a stop-gradient on the consistency side. Optimal results emerge when combining stop-gradients on both objectives with confidence weighting, confirming that asymmetric optimization coupled with confidence-aware transfer is crucial for stable latent learning.

Method	V*	HRBench4K	HRBench8K
Full Model	82.72	72.38	68.75
w/o Query Head	81.15	71.75	67.88
w/o SPI Module	80.25	70.50	66.25
w/ Indep. Codebooks	78.53	71.75	67.62
w/o Distance	81.68	71.00	67.25

Table 6: Ablation study of the Query Head and SPI Module design. We evaluate the effect of removing the Query Head, replacing the shared codebook with Independent Codebooks (Indep. Codebooks), and removing distance-based RBF matching.

Effect of the Query Head and Shared Prototype Interface. Table 6 ablates the structural designs of the latent interface. Removing the Query Head degrades performance, validating the need for branch-specific projections before codebook matching. Crucially, replacing the shared codebook with separate ones causes the most severe drop, underscoring that a unified prototype space is essential for effective prior-posterior coupling. Furthermore, discarding the RBF-based matching mechanism weakens results, proving the superiority of SPI over plain projection mappings. These findings confirm the indispensability of both the Query Head and the SPI framework.

Further ablations detailing stop-gradient optimization dynamics and token budget effects on spatial sensitivity are provided in Appendix E.

5 Conclusion

We introduce PEARL to advance latent reasoning in MLLMs. Rather than relying on predefined routes, PEARL enables the model to autonomously explore reasoning pathways from input to output, leveraging a Shared Prototype Interface and Reciprocal Divergence Constraints for stable distillation. Evaluations show PEARL achieves state-of-the-art performance, outperforming both discrete and latent baselines.

6 Limitations

A critical open challenge and limitation of our current framework is how to better adapt Explainable AI (XAI) methodologies to effectively interpret, probe, or verbalize continuous latent thinking. Developing novel XAI techniques specifically tailored to demystify these latent reasoning trajectories will be a vital direction for future research.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025a. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Sule Bai, Mingxing Li, Yong Liu, Jing Tang, Haoji Zhang, Lei Sun, Xiangxiang Chu, and Yansong Tang. 2025b. Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. *Preprint*, arXiv:2505.14231.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. 2025. Soft tokens, hard truths. *arXiv preprint arXiv:2509.19170*.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758.

Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. 2025. Mint-cot: Enabling interleaved visual tokens in mathematical chain-of-thought reasoning. *arXiv preprint arXiv:2506.05331*.

Zihui Cheng, Qiguang Chen, Xiao Xu, Jiaqi Wang, Weiyun Wang, Hao Fei, Yidong Wang, Alex Jinpeng Wang, Zhi Chen, Wanxiang Che, and 1 others. 2026. Visual thoughts: A unified perspective of understanding multimodal chain-of-thought. *Advances in Neural Information Processing Systems*, 38:96084–96112.

Jiwan Chung, Junhyeok Kim, Siyeol Kim, Jaeyoung Lee, Min Soo Kim, and Youngjae Yu. 2025. Don't look only once: Towards multimodal interactive reasoning with selective visual revisitation. *arXiv e-prints*, pages arXiv–2505.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Richard Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. 2025. Refocus: Visual editing as a chain of thought for structured image understanding. In *International Conference on Machine Learning*, pages 17783–17805. PMLR.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. *Preprint*, arXiv:2310.02226.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, and 1 others. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Minjie Hong, Zirun Guo, Yan Xia, Zehan Wang, Ziang Zhang, Tao Jin, and Zhou Zhao. 2025. Apo: Enhancing reasoning ability of mllms via asymmetric policy optimization. *arXiv preprint arXiv:2506.21655*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. 2023. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Fei zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2026. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *The Fourteenth International Conference on Learning Representations*.

Chaoya Jiang, Yongrui Heng, Wei Ye, Han Yang, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. 2025. Vlm-r^3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. *Preprint*, arXiv:2505.16192.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Ang Li, Charles L. Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. 2026a. Zebra-cot: A dataset for interleaved vision-language reasoning. In *The Fourteenth International Conference on Learning Representations*.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. 2026b. Latent visual reasoning. In *The Fourteenth International Conference on Learning Representations*.

Tianqianjin Lin, Xi Zhao, Xingyao Zhang, Rujiao Long, Yi Xu, Zhuoren Jiang, Wenbo Su, and Bo Zheng. 2025. Ravr: Reference-answer-guided variational reasoning for large language models. *arXiv preprint arXiv:2510.25206*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ryan Liu, Jiayi Geng, Addison J Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L Griffiths. 2024. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*.

Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. 2025. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. *Preprint*, arXiv:2505.19702.

Jacob Pfau, William Merrill, and Samuel R Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, and Jie Tang. 2025. Cogcom: A visual language model with chain-of-manipulations reasoning. In *The Thirteenth International Conference on Learning Representations*.

Arijit Ray, Ahmed Abdelkader, Chengzhi Mao, Bryan A. Plummer, Kate Saenko, Ranjay Krishna, Leonidas Guibas, and Wen-Sheng Chu. 2025. Mull-tokens: Modality-agnostic latent thinking. *Preprint*, arXiv:2512.10941.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael J Tarr, Aviral Kumar, and Katerina Fragkiadaki. 2025. Grounded reinforcement learning for visual reasoning. *arXiv preprint arXiv:2505.23678*.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual cot: Unleashing chain-of-thought reasoning in multi-modal language models. *arXiv preprint arXiv:2403.16999*, 2.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28.

Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. 2025. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *arXiv preprint arXiv:2504.10342*.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. 2026. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, and 1 others. 2025. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*.

Yuhta Takida, Takashi Shibuya, WeiHsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. 2022. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization. *arXiv preprint arXiv:2205.07547*.

Chameleon Team. 2025. Chameleon: Mixed-modal early-fusion foundation models. *Preprint*, arXiv:2405.09818.

Aaron Van Den Oord, Oriol Vinyals, and 1 others. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Fanmeng Wang, Haotian Liu, Guojiang Zhao, Hongteng Xu, and Zhifeng Gao. 2026a. Regular: Variational latent reasoning guided by rendered chain-of-thought. *Preprint*, arXiv:2601.23184.

Jianwei Wang, Ziming Wu, Fuming Lai, Shaobing Lian, and Ziqian Zeng. 2025a. Synadapt: Learning adaptive reasoning in large language models via synthetic continuous chain-of-thought. *arXiv preprint arXiv:2508.00574*.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. 2024. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Preprint*, arXiv:2406.14852.

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

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. 2025c. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru WANG, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and Heng Ji. 2026b. Perception-aware policy optimization for multimodal reasoning. In *The Fourteenth International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13084–13094. IEEE.

Zhongxing Xu, Zhonghua Wang, Zhe Qian, Dachuan Shi, Feilong Tang, Ming Hu, Shiyan Su, Xiaocheng Zou, Wei Feng, Dwarikanath Mahapatra, and 1 others. 2026. Thinking in uncertainty: Mitigating hallucinations in mlrms with latent entropy-aware decoding. *arXiv preprint arXiv:2603.13366*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and 1 others. 2025. Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl. *arXiv e-prints*, pages arXiv–2505.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and XingYu. 2026. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Xiangxin Zhou, Zichen Liu, Haonan Wang, Chao Du, Min Lin, Chongxuan Li, Liang Wang, and Tianyu Pang. 2025. Variational reasoning for language models. *arXiv preprint arXiv:2509.22637*.

A Main Notation Introduction

For clarity, Table 7 summarizes the main notations used throughout the paper.

B Theoretical Motivation of Reciprocal Divergence Constraints

In this section, we provide the theoretical motivation and probabilistic interpretation behind the Reciprocal Divergence Constraints introduced in Section 3.4. In particular, we explain why a bidirectional asymmetric formulation with stop-gradient is beneficial, and why the confidence weight w admits a principled evidence-based interpretation.

B.1 Necessity of Bidirectional Asymmetric Constraints

Our model involves two conditional latent distributions in the shared prototype space: the answer-aware posterior and the answer-agnostic prior. At each latent position i, these are characterized by the prototype assignments q_i and p_i, respectively. The central challenge is bridging the *train-test inference gap*, ensuring that the prior can successfully approximate the posterior's informative latent structures without collapsing to degenerate or overly uninformative latent assignments.

In standard variational learning, a forward KL term $\mathcal{D}_{\mathrm{KL}}(q_i \,\|\, p_i)$ is commonly used to regularize the approximate posterior toward the prior as part of the ELBO (Kingma and Welling, 2013). However, in our setting, both p_i and q_i are parameterized neural conditional distributions mapped to a discrete prototype space. Under this setup, using only a single directional divergence, or jointly optimizing a symmetric divergence without gradient blocking, can lead to undesirable optimization behavior:

- **Limitations of updating posterior toward prior alone:** If we exclusively penalize the divergence of the posterior q_i from the prior p_i, the posterior is discouraged from encoding answer-specific structures not yet predictable by the prior. Early in training, when the prior is uninformative and roughly uniform, unconditionally pulling the posterior toward it suppresses the inference branch's ability to utilize the target answer. This manifests as *posterior collapse*, where the posterior degrades to match an under-trained prior.

- **Limitations of updating prior toward posterior alone:** Conversely, forcing the prior p_i to unconditionally match the posterior q_i acts as a naive knowledge distillation (Hinton et al., 2015). Because the posterior has access to the target answer, its latent assignments are often highly sharp and deterministic. Forcing the answer-agnostic prior to universally match these sharp targets—especially on hard samples where it lacks the contextual information to deduce them—induces severe gradient instability and forces the prior into overconfident but incorrect latent assignments.

- **Instability of Joint Updates:** As demonstrated in self-supervised learning frameworks such as SimSiam (Chen and He, 2021) and BYOL (Grill et al., 2020), the stop-gradient operator is essential to prevent representation collapse when optimizing a distance metric between two parameterized branches; without it, co-adaptation typically leads to degenerate solutions, collapsing the discriminative structure of the shared prototype space.

To resolve this, we decouple the regularization into an alternating Teacher-Student projection mechanism using the $\mathrm{sg}(\cdot)$ operator. The stop-gradient turns each divergence term into a one-sided projection step, preventing simultaneous representation drift. The **Alignment objective** ($\mathcal{L}_{\mathrm{align}}$) updates only the posterior, acting as a structural bound that restricts posterior inference to the realizable support of the prior. Conversely, the **Consistency objective** ($\tilde{\mathcal{L}}_{\mathrm{consist}}$) updates only the prior, enabling stable distillation of informative assignments back to the prior.

B.2 Probabilistic Interpretation of the Confidence Weight w

While the unweighted consistency objective $\tilde{\mathcal{L}}_{\mathrm{consist}} = \frac{1}{k}\sum_{i=1}^{k} \mathcal{D}_{\mathrm{KL}}(p_i \,\|\, \mathrm{sg}(q_i))$ provides a correct distillation direction, directly optimizing it assumes that the posterior target q_i is always a reliable supervision signal for the prior. This assumption often fails on difficult samples where the prior lacks the contextual capacity to explain the target answer, leading to spurious gradients that force the prior to overfit to unachievable targets.

To alleviate this, we modulate the consistency loss with $w = \mathrm{sg}\big(\exp(-\bar{\mathcal{L}}^{p}_{\mathrm{rec}})\big)$. We formalize the theoretical soundness of this choice from the perspectives of evidence-weighted risk minimization and self-paced learning.

Notation	Type	Meaning
I, T	input context	Multimodal input consisting of an image I and a textual prompt T.
$a_1, \ldots, a_m$	sequence	Target output sequence representing the m answer tokens.
$a_{<t}$	sequence prefix	Target answer prefix before time step t.
$L_1, \ldots, L_k$	sequence	Placeholder tokens inserted for latent reasoning.
$b \in \{p, q\}$	indicator	Branch indicator denoting the prior (p) or posterior (q) branch.
$\mathbf{X}^b$	sequence	Unified input sequence constructed for branch b.
$\mathbf{H}_L^b$	$\mathbb{R}^{k \times d}$	Hidden states extracted at the k latent positions from the MLLM backbone.
$\mathbf{Z}^b = [z_1^b, \ldots, z_k^b]$	$\mathbb{R}^{k \times d_z}$	Latent queries mapped by the branch-specific Query Head.
z_i^b	$\mathbb{R}^{d_z}$	The continuous latent query at the i-th latent slot for branch b.
$\mathcal{E} = \{e_1, \ldots, e_Q\}$	codebook	Shared prototype codebook containing Q prototypes.
e_j	$\mathbb{R}^{d_z}$	The j-th learnable prototype vector in the codebook.
$\boldsymbol{\alpha}_i^b$	$\mathbb{R}^Q$	Soft assignment weights (distribution) over the codebook for the i-th latent query.
z'^b_i	$\mathbb{R}^{d_z}$	Soft aggregated prototype for the i-th latent position.
$\mathbf{L}'^b = [L'^b_1, \ldots, L'^b_k]$	$\mathbb{R}^{k \times d}$	Refined latent embeddings projected back to the decoder hidden space.
p_i, q_i	Δ^Q	Prior and posterior prototype distributions at the i-th latent position.
w	scalar	Evidence-guided confidence weight derived from prior reconstruction quality.
$\mathcal{L}_{\text{align}}$	scalar	Alignment objective regularizing the posterior toward the prior.
$\mathcal{L}_{\text{consist}}$	scalar	Weighted consistency objective encouraging the prior to match the posterior.
$\mathcal{L}_{\text{rec}}$	scalar	Standard next-token prediction (reconstruction) loss.

Table 7: Meanings of the main notations used in the paper.

1. Geometric Mean Likelihood and Evidence Estimation. Recall that the prior reconstruction loss is defined as the average negative log-likelihood (NLL) of the target answer sequence $A = (a_1, \ldots, a_m)$ under the prior latent sequence $\mathbf{L}'^p$:

$$\bar{\mathcal{L}}_{\text{rec}}^p = -\frac{1}{m} \sum_{t=1}^{m} \log P(a_t \mid a_{<t}, \mathbf{X}_{\text{ctx}}, \mathbf{L}'^p). \quad (20)$$

By exponentiating the negative NLL, the pre-stop-gradient value of w mathematically corresponds to the geometric mean likelihood of the target answer:

$$\exp\left(-\bar{\mathcal{L}}_{\text{rec}}^p\right) = \left(\prod_{t=1}^{m} P_\theta(a_t \mid a_{<t}, \mathbf{X}_{\text{ctx}}, \mathbf{L}'^p)\right)^{\frac{1}{m}}. \quad (21)$$

Since $P_\theta(a_t \mid \ldots) \in (0, 1]$, we naturally have $w \in (0, 1]$. More importantly, w acts as a direct, probabilistic measure of *generative evidence*: it quantifies how well the prior's current latent state can explain the target answer.

2. Evidence-Weighted Distillation Risk. Why can generative evidence serve as a proxy for the reliability of the posterior target? Intuitively, if the prior can already reconstruct the answer reasonably well (yielding a high weight w), its latent manifold is likely close to that of the posterior, making q_i a more achievable and reliable target for distillation. Conversely, a low w suggests that the posterior may lie in a region that is currently difficult for the prior to realize, so enforcing strict alignment may introduce spurious optimization signals.

To formalize this intuition, let $C \in \{0, 1\}$ denote a latent reliability variable indicating whether the posterior target is achievable by the prior for a given sample. Since C is unobserved, we use the prior's current generative evidence as a tractable surrogate for its achievability probability, and model

$$P(C = 1 \mid x, A) \propto w(x, A),$$

where $w(x, A)$ is defined from the prior reconstruction loss. Treating $\tilde{\mathcal{L}}_{\text{consist}}$ as the observed dis-

tillation loss for sample (x, A) and C as an unobserved reliability gate, the expected gated loss becomes $\mathbb{E}_C\left[\tilde{\mathcal{L}}_{\text{consist}} \cdot \mathbf{1}_{\{C=1\}}\right] = P(C = 1 \mid x, A)\, \tilde{\mathcal{L}}_{\text{consist}} \propto w(x, A)\, \tilde{\mathcal{L}}_{\text{consist}}$. This motivates the following evidence-weighted empirical risk, denoted as $\tilde{\mathcal{R}}_{\text{distill}}(\theta)$:

$$\tilde{\mathcal{R}}_{\text{distill}}(\theta) = \mathbb{E}_{(x,A)\sim\mathcal{D}}\Bigg[w(x, A) \cdot \frac{1}{k} \sum_{i=1}^{k} \mathcal{D}_{\text{KL}}\Big(p_i(x) \,\|\, \text{sg}(q_i(x, A))\Big)\Bigg]. \tag{22}$$

Importantly, because the weight is defined as $w = \exp(-\bar{\mathcal{L}}^p_{\text{rec}})$, it becomes small for samples on which the prior incurs high reconstruction loss. Since w is treated as a stop-gradient scalar during backpropagation, the KL-induced gradient for each sample is directly rescaled by w. As a result, samples whose posterior targets are less likely to be achievable are automatically downweighted, reducing the influence of potentially unreliable distillation signals and improving optimization stability.

Remark 1 (Adaptive Gradient Attenuation). Let $g_j = \nabla_\theta \tilde{\mathcal{L}}^{(j)}_{\text{consist}}$ denote the unweighted gradient for sample j. The weighted estimator becomes $g^w_j = w_j g_j$, where $w_j \in (0, 1]$. Because w_j decreases exponentially with the prior reconstruction loss, samples with weaker generative evidence contribute less to the parameter update. This mechanism adaptively attenuates gradients from samples whose posterior targets are less likely to be achievable, thereby stabilizing optimization without requiring hard gradient clipping.

3. Implicit Curriculum vs. Hard-Example Mining. To intuitively understand the necessity of our evidence weight w, it is instructive to contrast it with hard-example mining strategies, such as Focal Loss (Lin et al., 2017).

Our weight $w = \exp(-\bar{\mathcal{L}}^p_{\text{rec}})$ mathematically represents the likelihood of the target answer under the prior. By assigning higher weights to samples with lower prior reconstruction loss, our mechanism acts as an *implicit curriculum* (Bengio et al., 2009), dynamically pacing the learning process based on the prior's evolving capacity. Early in training, it strategically shields the prior from overly complex, unachievable posterior targets. Only when the prior develops sufficient contextual capacity (thereby lowering the reconstruction loss and increasing w) does it progressively learn from harder samples.

Method	V*	HRBench4K	HRBench8K
Focal Loss-style	78.53	71.12	67.38
Uniform Weighting	80.63	72.00	67.88
Ours	**82.72**	**72.38**	**68.75**

Table 8: **Ablation on consistency weighting strategies.** Replacing our curriculum-based weight (w) with uniform weighting ($w = 1$) or a Focal Loss-style strategy ($1 - w$) degrades performance, demonstrating that aggressively matching unachievable targets disrupts distillation.

In contrast, a standard hard-example mining approach behaves oppositely. To illustrate this, we define a **Focal Loss-style weighting** mechanism as $w^{\text{focal}} = 1 - w$. Since w represents the likelihood of the target answer, $1-w$ exactly mirrors the Focal Loss modulating factor $(1 - p_t)^\gamma$ with $\gamma = 1$. This strategy aggressively up-weights difficult samples where the prior struggles the most.

Empirical Validation. We validate this intuition in Table 8. As shown, substituting our evidence-guided weight w with the Focal Loss-style weight $(1 - w)$ leads to a severe performance degradation across all benchmarks (e.g., a striking 4.19% drop on V*). While hard-example mining is highly effective for refining classification boundaries, it proves detrimental in our bidirectional distillation setting. Forcing the answer-agnostic prior to blindly fit unachievable, high-loss posterior targets amplifies spurious gradients and disrupts stable representation learning. Furthermore, omitting the weight entirely (Uniform Weighting, $w = 1$) also yields suboptimal results. These findings empirically confirm our core design: adaptive gradient attenuation via w is critical for effectively coupling the prior and posterior spaces.

B.3 Geometric and Optimization Dynamics of the Stop-Gradient in Prototype Space

This section expands on the theoretical justification for imposing the stop-gradient on the second argument of the KL divergences in Eqs. (16) and (17), contrasting the optimization landscape with alternative formulations.

Mode-Seeking Geometry vs. Representation Collapse The choice to optimize $\mathcal{D}_{\text{KL}}(P \,\|\, \text{sg}(Q))$ instead of a joint optimization $\mathcal{D}_{\text{KL}}(P \,\|\, Q)$ is critical for maintaining a meaningful Shared Prototype Interface (SPI). If both distributions were opti-

mized simultaneously without a stop-gradient, the neural network would naturally seek the path of least resistance to minimize the loss to zero. In a discrete prototype space, this typically induces **representation collapse**, where both the prior and posterior lazily collapse to predicting a single, uninformative prototype (e.g., assigning 100% probability to Prototype #1 for every query), completely disregarding the multimodal context. The stop-gradient breaks this symmetry, forcing one network to align with an immutable anchor, thereby preserving the rich, query-dependent diversity of the prototype assignments.

Support Containment in Discrete Spaces. The mathematical asymmetry of the KL divergence, defined as $\sum P(x) \log \frac{P(x)}{Q(x)}$, means that optimizing its first argument behaves fundamentally differently than optimizing its second argument.

By formulating our losses as $\mathcal{D}_{\mathrm{KL}}(P \,\|\, \mathrm{sg}(Q))$, the optimized distribution P dictates the weighting of the logarithm. If P assigns probability mass to a prototype where the fixed target Q has near-zero mass, the ratio $\frac{P(x)}{Q(x)}$ explodes, resulting in an infinite penalty. This creates a strictly **mode-seeking** effect. For discrete categorical distributions over the prototype codebook, this mathematical property guarantees that the optimized distribution safely restricts its probability mass to the exact subset of prototypes validated by the fixed target. It effectively eliminates the destructive "mass-covering" gradient variance that would occur if the arguments were inverted.

C Experiment Setup

C.1 Evaluation Benchmarks Details

To provide a comprehensive assessment of our model's capabilities, we carefully select benchmarks that target specific bottlenecks in current large vision-language models (LVLMs):

- **V* (Wu and Xie, 2024):** This benchmark is specifically designed to evaluate visual search and fine-grained perception capabilities. It tests whether the model can accurately locate and process small objects or intricate details within complex, high-resolution visual scenes, which is often a weakness in models heavily reliant on extreme visual downsampling.

- **HRBench4K and HRBench8K (Wang et al., 2025c):** To push the limits of high-resolution image processing, we utilize HRBench4K and 8K. These datasets evaluate the model's capacity for dense feature extraction and document-level understanding at ultra-high resolutions (up to 4K and 8K), ensuring that our method effectively preserves fine visual details without critical information loss.

- **BLINK (Fu et al., 2024):** Unlike traditional VQA benchmarks, BLINK focuses on core visual cognition tasks that humans can solve effortlessly ("in a blink") but remain highly challenging for multimodal models. We utilize it to rigorously test our model's complex visual reasoning, including spatial logic, relative depth perception, and overlapping object recognition.

- **VisualPuzzles (Song et al., 2025):** To evaluate robustness beyond natural image distributions, we employ the abstract reasoning categories of VisualPuzzles. This benchmark requires models to discover underlying logical patterns (e.g., matrix reasoning and geometric transformations), providing a reliable metric for our model's out-of-distribution (OOD) generalization and abstract cognitive abilities.

C.2 Baselines

We benchmark our approach against state-of-the-art MLLMs across two dominant paradigms: *discrete reasoning* and *latent reasoning*.

Discrete Reasoning. These models enhance multi-step logical deduction by generating explicit textual trajectories or leveraging external tools, typically optimized via reinforcement learning (RL).

- **Vision-R1 (Huang et al., 2026):** Employs RL to elicit explicit "think-before-answering" chains-of-thought (CoT), heavily optimizing the visual step-by-step reasoning process.

- **PAPO (Wang et al., 2026b):** Regularizes RL-based reasoning with an auxiliary perception loss, mitigating hallucinations by ensuring intermediate steps remain strictly grounded in visual features.

- **PixelReasoner (Su et al., 2026):** An interactive paradigm that iteratively manipulates or refines the input image via external tools to resolve fine-grained visual ambiguities during reasoning.

- **DeepEyes (Zheng et al., 2026):** Directly embeds multi-tool execution (e.g., grounding and web search) into the RL policy, enabling an end-to-end, agentic visual problem-solving workflow.

Latent Reasoning. These methods bypass explicit textual CoT, executing intermediate computational steps directly within the continuous embedding space for greater efficiency.

- **LVR (Li et al., 2026b):** Interleaves token generation with the autoregressive reconstruction of "latent visual thoughts," processing continuous visual semantics without materializing them into discrete text.

- **Mull-Tokens (Ray et al., 2025):** Allocates modality-agnostic latent tokens to serve as a continuous internal scratchpad, allowing the model to accumulate and store intermediate compute before generating the final response.

C.3 Implementation Details

Base Model and Processor. We instantiate our framework using Qwen2.5-VL-7B-Instruct as the base MLLM, retaining its official processor and tokenizer. To facilitate continuous reasoning, we expand the tokenizer's vocabulary with latent-specific special tokens: a start token, an end token, and a series of placeholder tokens designed to allocate latent slots within the autoregressive input stream.

Training Data Mixture. We curate a comprehensive multimodal mixture to ensure the learned latent space generalizes across a broad spectrum of cognitive tasks. The training corpus integrates the following datasets:

- **Visual-CoT (Shao et al., 2024):** Serves as the foundational corpus for step-by-step reasoning, training the model to decompose complex queries and bridge high-level semantics with visual evidence.

- **ReFocus (Fu et al., 2025):** Enforces precise spatial grounding, requiring the model to maintain focus on fine-grained visual details and specific regions of interest.

- **CogCoM (Qi et al., 2025):** Enhances multi-hop deductive capabilities by challenging the model to synthesize intricate visual and textual cues into valid logical conclusions.

- **Zebra-CoT (Li et al., 2026a):** Strengthens compositional and relational reasoning among multiple objects, mitigating the model's reliance on superficial data correlations.

Data Formatting and Optimization. To ensure consistency, we convert all training data into a standard multimodal conversational structure. A rigorous filtering step is applied to remove any examples with truncated assistant replies. For the retained data, we automatically inject a sequence of latent tokens directly after the assistant's response prefix. The entire training process takes roughly 10 hours using a cluster of 32 NVIDIA A100 GPUs.

Optimization details. We train the model with AdamW using fused PyTorch implementation, bf16 mixed precision, gradient checkpointing, cosine learning-rate scheduling, and a warmup ratio of 0.05. The vision encoder is frozen throughout training, while the language backbone and our modules are jointly optimized. We also enable TF32 matrix multiplication where available for training acceleration. The exact batch size, gradient accumulation steps, learning rate, and training epochs follow the settings reported in the main experiments.

Embedding initialization and freezing strategy. To accommodate the continuous reasoning process, we expand the tokenizer's vocabulary with latent-specific special tokens (including start/end markers and the latent placeholders $L_1, \ldots, L_k$) and resize the model vocabulary accordingly. The embeddings for these newly introduced tokens are initialized using the mean of the pretrained embedding matrix.

D Out-of-Distribution Generalization on VisualPuzzles

Table 9 reports the performance on the out-of-distribution (OOD) VisualPuzzles benchmark. Overall, our method achieves the highest average accuracy (33.13). It particularly excels in **deductive** (51.00, +3.50 over Qwen2.5-VL-7B), **spatial** (30.77), and **inductive** (29.19) reasoning. These consistent gains indicate that our approach effectively preserves structural visual information and robustly captures logical patterns in unfamiliar settings.

While baselines like DeepEyes and Qwen2.5-VL-7B retain an edge in **algorithmic** and **analogical** categories—likely benefiting from stronger explicit procedural priors—our method delivers the most balanced performance across all sub-tasks. Rather than overfitting to specific reasoning patterns, our reciprocal latent framework successfully acquires a more generalizable multimodal reasoning capability under distribution shifts.

Method	Overall	Algorithmic	Analogical	Deductive	Inductive	Spatial
Qwen2.5-VL-7B	32.71	37.02	21.80	47.50	26.32	21.80
Pangea-7B	31.30	32.40	23.70	38.50	28.70	32.50
Deepeyes	32.96	37.79	27.01	41.00	26.79	27.01
LVR	27.74	28.63	23.22	36.00	28.23	24.12
LLaVA-OneVision-72B	30.80	34.70	26.50	37.00	27.30	28.70
Ours	33.13	33.21	23.22	51.00	29.19	30.77

Table 9: Performance on the out-of-distribution VisualPuzzles benchmark.

$\mathrm{sg}(p)$	$\mathrm{sg}(q)$	V*	HRBench4K	HRBench8K
×	×	81.68	72.38	68.25
✓	×	30.37	59.13	59.38
×	✓	76.96	71.38	67.88

Table 10: Ablation study on the position of the stop-gradient (sg) operator in $\mathcal{L}_{\text{align}}$

$\mathrm{sg}(p)$	$\mathrm{sg}(q)$	V*	HRBench4K	HRBench8K
×	×	81.15	71.50	67.88
✓	×	41.36	59.38	60.12
×	✓	76.96	71.38	66.75

Table 11: Ablation study on the position of the stop-gradient (sg) operator in $\mathcal{L}_{\text{consist}}$

E More Ablation Study

E.1 Reciprocal Divergence Constraints

To further dissect the optimization dynamics, we isolate the two divergence constraints ($\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{consist}}$). In Table 10 and Table 11, the model is trained with the standard reconstruction loss combined with *only one* constraint at a time.

The most striking phenomenon in both tables is the severe performance degradation in the second row, where $\mathrm{sg}(p)$ is applied. Under this setting, V* scores plunge drastically (e.g., to 30.37 and 41.36). Since the context-only prior p is the exact branch used during inference, blocking its gradient flow from the divergence constraint prevents it from updating its latent space to capture the reasoning process. Without learning this structure, the prior fails to generate meaningful latents, leading to a complete breakdown at inference time.

When applying $\mathrm{sg}(q)$ (the third row), the prior p is forced to unilaterally match a frozen, answer-aware posterior. While this avoids complete collapse, it still results in a noticeable performance drop (e.g., yielding 76.96 on V*). Because the posterior q has full access to the target answer, it easily encodes answer-specific shortcuts. Forcing the context-only prior to blindly predict these unachievable, answer-leaked representations overburdens the prior and misguides its training, thereby degrading its generalizable reasoning capability.

Removing the stop-gradient entirely $(\times, \times)$ avoids collapse but causes mutual adaptation, diluting the posterior's informativeness to meet the under-trained prior in the middle. These isolated failures validate our reciprocal design: a single unconstrained loss is insufficient. We must decouple the objectives by using $\mathcal{L}_{\text{align}}$ to anchor q to the support of $\mathrm{sg}(p)$, while using the weighted $\mathcal{L}_{\text{consist}}$ to safely distill $\mathrm{sg}(q)$ into p, preventing both inference collapse and posterior dilution.

E.2 Effect of the Number of Latent Tokens on Spatial Sensitivity.

To examine how the latent-token budget affects visual grounding, we visualize token-level sensitivity maps under different numbers of latent tokens ($k = 4, 8, 16$) on the same example. Figure 3 shows the patch-level perturbation sensitivity for the question: *"What word is written on the hang glider?"* Warmer colors indicate larger representation changes under perturbation.

Different token budgets lead to clearly different sensitivity patterns. With 4 latent tokens, the responses are relatively coarse and spread over surrounding background regions, indicating limited capacity for localized reasoning. With 8 latent tokens, the sensitivity maps become more concentrated around the small, question-relevant target while suppressing irrelevant areas such as the sky and cliffs, suggesting a better balance between capacity and focus. When the token number is increased to 16, the sensitivity patterns become more diffuse across tokens, and several tokens respond to broader background regions, implying increased

redundancy and weaker specialization. Overall, these visualizations are consistent with the quantitative ablation in the main text: a moderate token budget, especially $k = 8$, yields the most focused and task-relevant latent reasoning behavior.

F Qualitative Analysis of Latent Tokens

To investigate PEARL's internal representations during inference, we visualize token-level relevance heatmaps by measuring the sensitivity of each latent token L_i to patch-level image masking (where warmer colors indicate higher sensitivity). Figures 4–6 present representative examples across V*, HRBench4K, and HRBench8K, covering diverse reasoning tasks.

We observe that latent tokens consistently align with task-relevant image regions rather than visually salient but irrelevant backgrounds. For instance, tokens sharply attend to a small, low-contrast subject in the attribute recognition task (Figure 5), and tightly localize to a vehicle's rear panel while suppressing the expansive mountain background in text spotting (Figure 4 and Figure 6). This spatial selectivity confirms that latent tokens dynamically form task-conditioned representations grounded in question semantics, rather than extracting generic visual features.

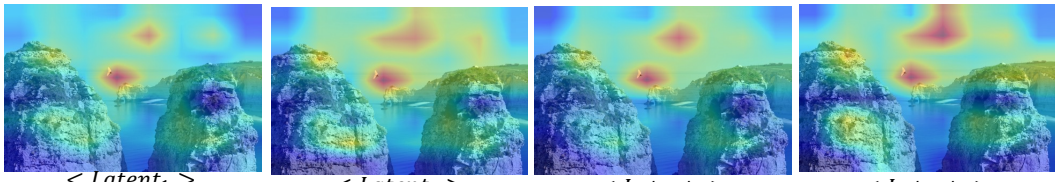
For questions requiring joint reasoning across multiple regions, the tokens exhibit complementary, non-redundant patterns. In the geographic reasoning example (Figure 5), some tokens (L_2, L_6) broadly scan candidate locations, while others (L_4, L_8) concentrate on distinct sub-regions, leaving irrelevant areas inactive. This division of labor naturally mirrors the *enumerate-then-compare* paradigm of explicit Chain-of-Thought (CoT) traces, executed entirely within the continuous latent space.

Crucially, this spatial concentration provides compelling evidence against answer leakage. If the tokens were merely memorizing target answers, their spatial sensitivity would appear diffuse and uncorrelated with the visual input. Instead, the sharply localized heatmaps confirm that latent tokens encode intermediate visual reasoning states—specifically, *where to look* and *what to compare*—rather than exploiting textual shortcuts. These findings empirically validate that our Reciprocal Divergence Constraints ($\mathcal{L}_{\mathrm{RDC}}$) effectively prevent the posterior branch from collapsing into ungeneralizable, answer-specific modes.

Question: What word is written on the hang glider?

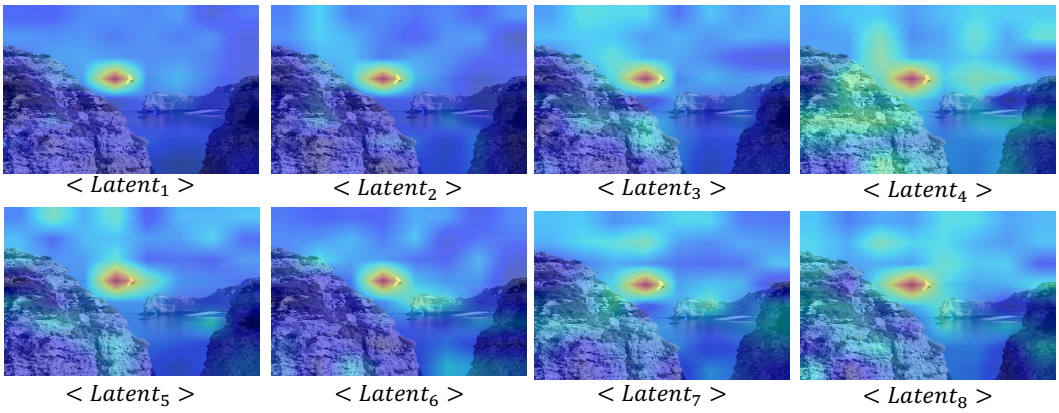
Question Image:

Heatmaps of 4 Tokens:



$< Latent_1 >$ $< Latent_2 >$ $< Latent_3 >$ $< Latent_4 >$

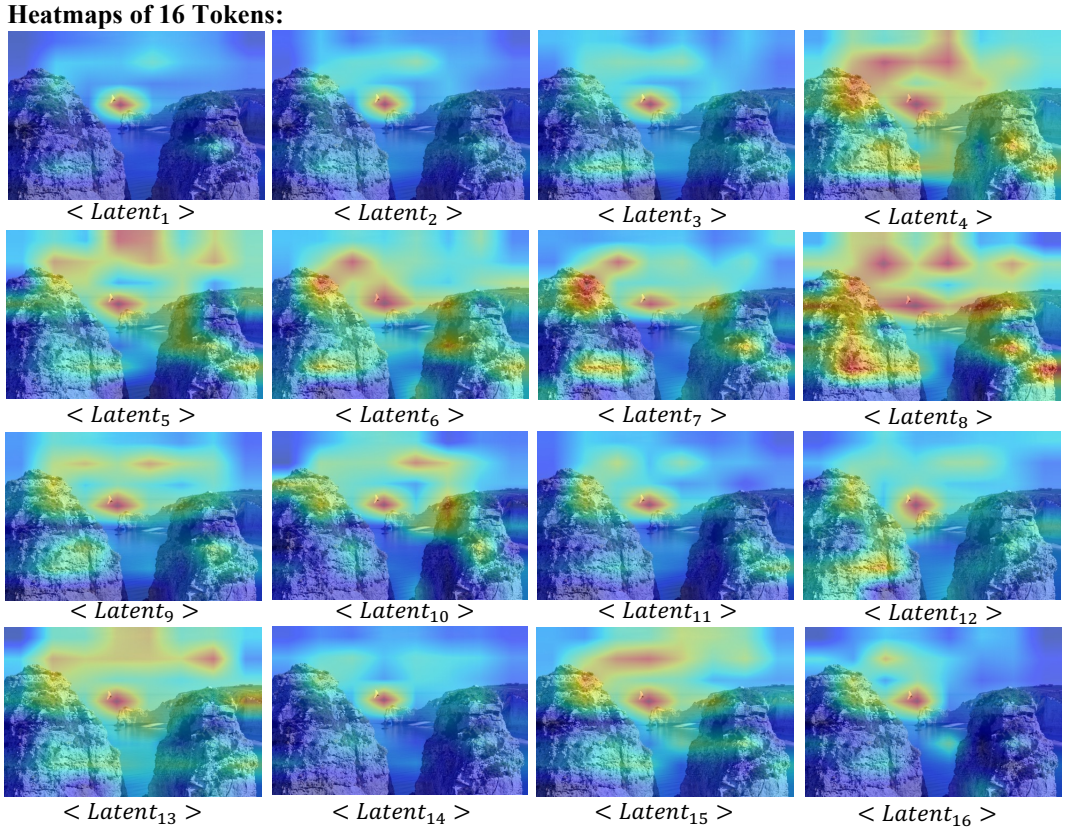
Heatmaps of 8 Tokens:



$< Latent_1 >$ $< Latent_2 >$ $< Latent_3 >$ $< Latent_4 >$

$< Latent_5 >$ $< Latent_6 >$ $< Latent_7 >$ $< Latent_8 >$

Heatmaps of 16 Tokens:



$< Latent_1 >$ $< Latent_2 >$ $< Latent_3 >$ $< Latent_4 >$

$< Latent_5 >$ $< Latent_6 >$ $< Latent_7 >$ $< Latent_8 >$

$< Latent_9 >$ $< Latent_{10} >$ $< Latent_{11} >$ $< Latent_{12} >$

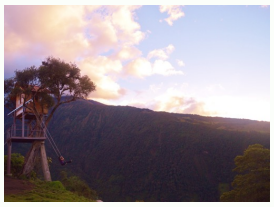
$< Latent_{13} >$ $< Latent_{14} >$ $< Latent_{15} >$ $< Latent_{16} >$

Figure 3: Effect of the number of latent reasoning tokens on patch-level perturbation sensitivity

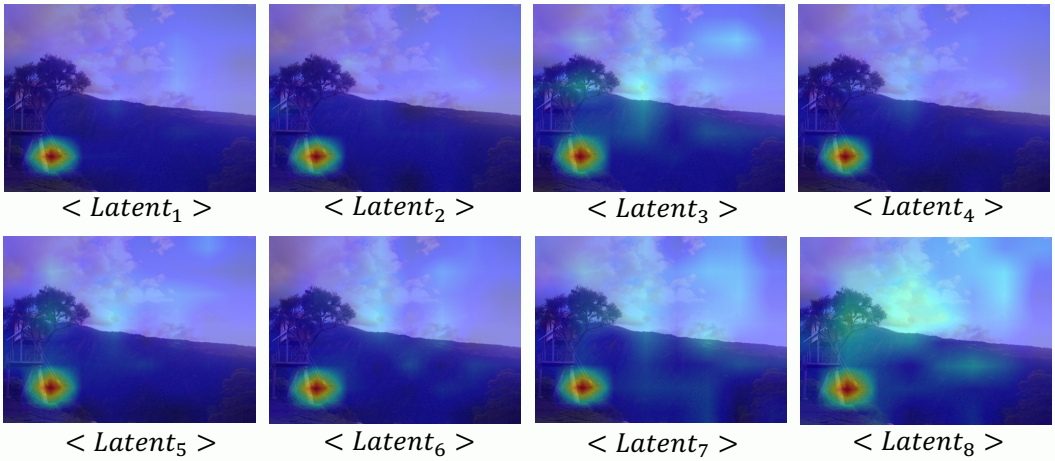
Examples of patch-level relevance heatmap

Question: What is the color of the woman's shirt?

Question Image:



Heatmaps:



Question: What is the color of the man's helmet?

Question Image:

Heatmaps:

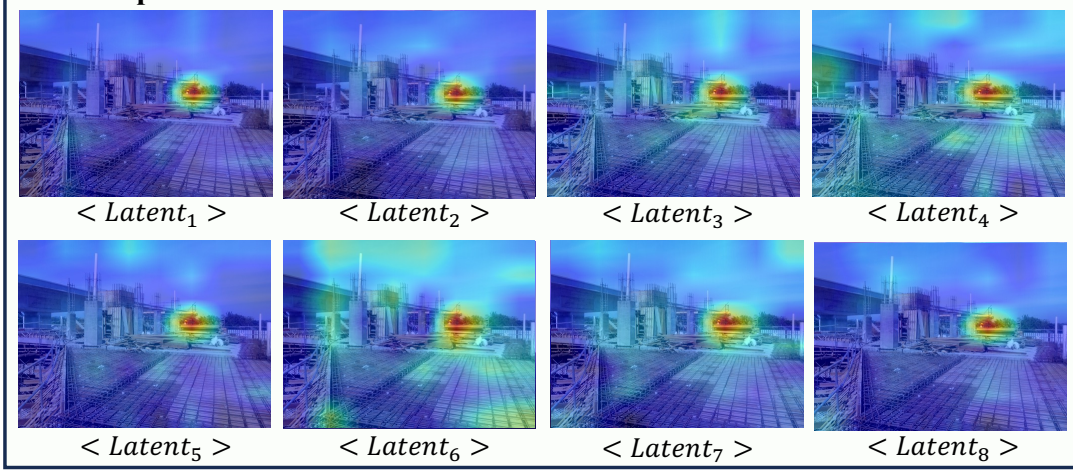


Figure 4: Sensitivity of latent reasoning tokens to patch-level image perturbations on the V*.

Examples of patch-level relevance heatmap

Question: What is the primary function of the object in the red circle?

Question Image:

Heatmaps:

Question: On the provided map, which two numbered locations belong to the same country?

Question Image:

Heatmaps:

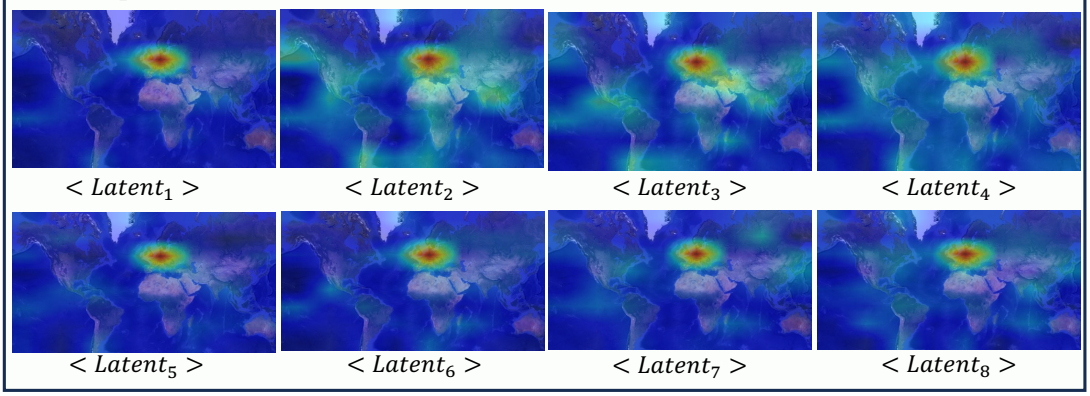


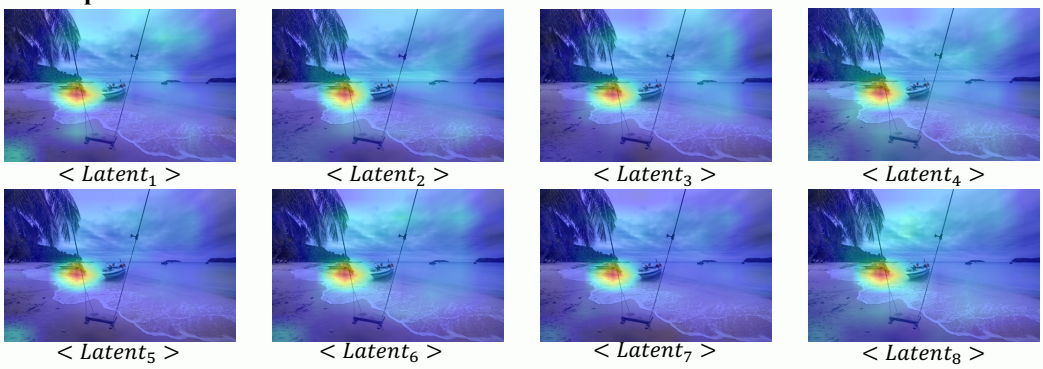
Figure 5: Sensitivity of latent reasoning tokens to patch-level image perturbations on the HRBench4K.

Examples of patch-level relevance heatmap

Question: What's the phone number written on the boat?

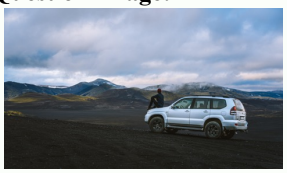
Question Image:

Heatmaps:



Question: What is the license plate number of the car?

Question Image:



Heatmaps:

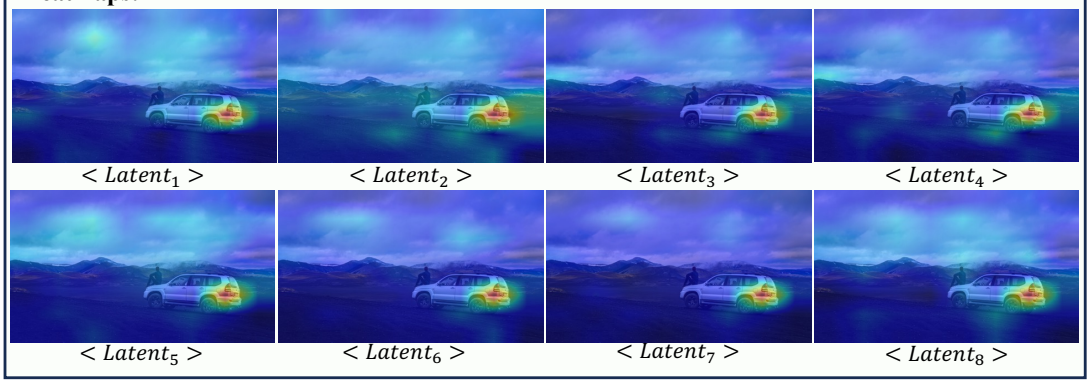


Figure 6: Sensitivity of latent reasoning tokens to patch-level image perturbations on the HRBench8K.