Hide to See: Reasoning-prefix Masking for Visual-anchored Thinking in VLM Distillation

Seonghoon Yu[1] **Dongjun Nam**[3] **Byung-Kwan Lee**[2,†] **Jeany Son**[3,†]
[1]KAIST [2]NVIDIA [3]POSTECH
seonghoon.yu@kaist.ac.kr byungkwanl@nvidia.com {june6423, jeany}@postech.ac.kr
Project Page: https://seonghoon-yu.github.io/Masking-KD-Page

Abstract

Recent think-answer approaches in VLMs, such as Qwen3-VL-Thinking, boost reasoning performance by leveraging intermediate thinking steps before the final answer, but their computational cost becomes substantial, especially for larger VLMs. To distill such capabilities into compact think-answer VLMs, a primary objective is to improve the student's ability to utilize visual evidence throughout its reasoning trace, as long think-answer traces suffer from visual forgetting issues. To this end, we introduce a novel think-answer distillation framework that encourages the student to anchor its thinking on visual information by masking the student's salient reasoning prefixes. To compensate for such masked textual cues, the student is encouraged to rely more on visual evidence as an alternative source of information during distillation. Our masking strategies include: 1) *token-wise salient reasoning-prefix masking*, which masks high-influence reasoning prefixes selectively for each next-token prediction, and 2) *self-paced masking budget scheduling*, which gradually increases the masking scale according to distillation difficulty, measured by the discrepancy between teacher–student distributions. In the distillation phase, the student is guided by our salient reasoning-prefix mask, which blocks both future tokens and salient reasoning cues, in place of the standard causal mask used for auto-regressive language modeling. Experimental results show that our approach outperforms recent open-source VLMs, VLM distillation, and self-distillation methods on multimodal reasoning benchmarks, while further analyzes confirm enhanced visual utilization along the student thinking process.

1 Introduction

Recent think-answer approaches in vision-language models (VLMs) and large language models (LLMs), such as ChatGPT 5.4 [22], Gemini 3.1 Pro [26], and DeepSeek-R1 [8], have achieved strong reasoning performance by explicitly generating reasoning before producing final answers. This paradigm has been widely adopted in subsequent VLMs [36, 29, 32, 15, 40] to improve reasoning ability, particularly in complex problem-solving tasks such as mathematical reasoning [38, 17] and scientific reasoning [37, 19]. However, the computational cost is particularly high for larger VLMs, limiting their deployment in resource-constrained scenarios such as on-device applications.

Knowledge distillation (KD) has emerged as a practical approach for reducing the computational overhead of large VLMs by transferring their capabilities to compact student models. Since the distinctive capability of VLMs lies in connecting visual inputs with language outputs, effective distillation should preserve the student's reliance on visual evidence when producing language predictions. Existing methods [4, 14, 6, 25, 7] mainly address this goal by transferring the teacher's visual knowledge through visual attention maps [14], vision token relations [4], or vision projector alignment [7]. While effective, these objectives primarily encourage the student to mimic the teacher's internal visual patterns, rather than guiding the student to anchor its own reasoning process in visual

† Corresponding authors.

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

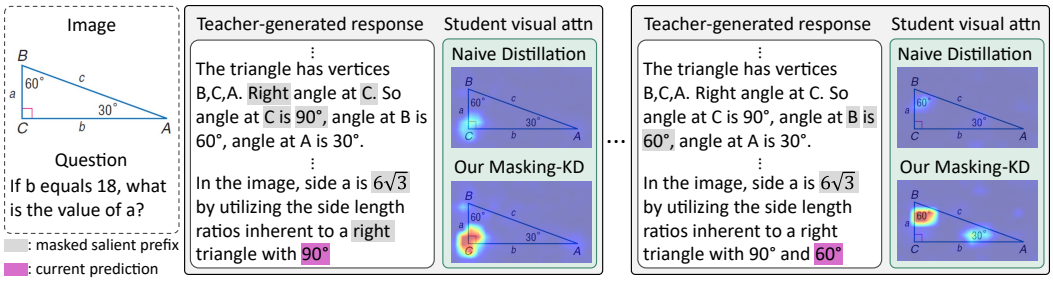


Figure 1: The illustration of our reasoning-prefix masking during VLM distillation. With full context of teacher's thinking trace (*i.e.*, naïve distillation), the student relies heavily on exposed textual prefixes to predict the current token, resulting in weak visual attention. In contrast, with masked salient reasoning prefixes (*i.e.*, our Masking-KD), the student exploits more visual evidence to compensate for missing textual reasoning cues, improving its visual-anchored thinking.

evidence. This gap becomes particularly problematic for think-answer VLMs, where long think-answer trajectories expose rich reasoning cues that themselves provide enough information to predict subsequent tokens (analyzed in Sec. 3.4), thereby diminishing the necessity to maintain sufficient visual reference throughout the thinking process, leading to visual forgetting [27, 32, 30].

In particular, when distilling such long traces of think-answer VLMs, the student relies heavily on a small set of exposed textual cues, which receive disproportionately high attention values (Fig. 2), at every decoding step. This suggests that only a few reasoning cues already provide sufficient information to follow the teacher's think-answer trace, which may reduce the student's need to learn from visual input. This motivates our key question: *If salient reasoning prefixes allow the student to imitate the teacher with less reliance on the image, can masking such prefixes encourage the student to exploit more visual evidence as an alternative to the masked salient textual cues?*

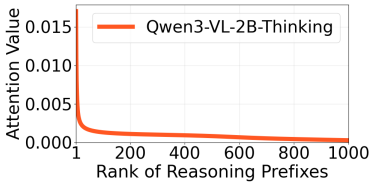


Figure 2: Reliance on salient cues

In this paper, we introduce Masking-KD, a novel distillation framework for think-answer models that enhances the student's ability to anchor its thinking in visual evidence by masking the student's prefixed reasoning cues, thereby encouraging it to rely more on visual cues to compensate for missing textual evidence (Fig. 1). Our salient reasoning-prefix mask is carefully constructed via: 1) **token-wise salient reasoning-prefix masking**, which identifies and masks the high-attention (*i.e.*, salient) reasoning-context tokens selectively for each next-token prediction, since the most influential contextual cues vary across decoding steps, and masking them in a token-wise different manner; and 2) **self-paced masking budget scheduling**, adaptively adjusts the masking scale for each next-token prediction according to its teacher-student KL divergence, assigning stronger masking to tokens that are easily imitated to amplify weak learning signals. During distillation, the student is guided by our salient reasoning-prefix mask, which limits access to both future tokens and salient reasoning cues, instead of using the standard causal mask in auto-regressive language modeling.

In our experiments, the proposed framework outperforms recent open-source VLMs and VLM distillation methods on multimodal reasoning benchmarks. It also demonstrates effectiveness in self-distillation, where the student serves as its own teacher rather than a stronger teacher. Furthermore, our analysis shows that masking salient reasoning prefixes during distillation improves the student's ability to derive its thinking from visual evidence. Our contributions are summarized as follows:

- We introduce Masking-KD, a novel think-answer distillation framework that enhances the student's ability to ground its thinking in visual evidence by masking the student's prefixed reasoning-context, encouraging the student to draw more on visual sources to compensate for missing textual cues.
- We propose *token-wise salient reasoning-prefix masking* that identifies and masks high-influence prefixed reasoning cues selectively for each next-token prediction, reflecting that the most influential reasoning-context varies across every decoding step and masking them token-wise differently.
- We present *self-paced masking budget scheduling* that adaptively adjusts the masking budget for each next-token prediction based on its teacher-student KL divergence, applying more aggressive masking on easily imitated tokens for the student so as to strengthen weak distillation signals.

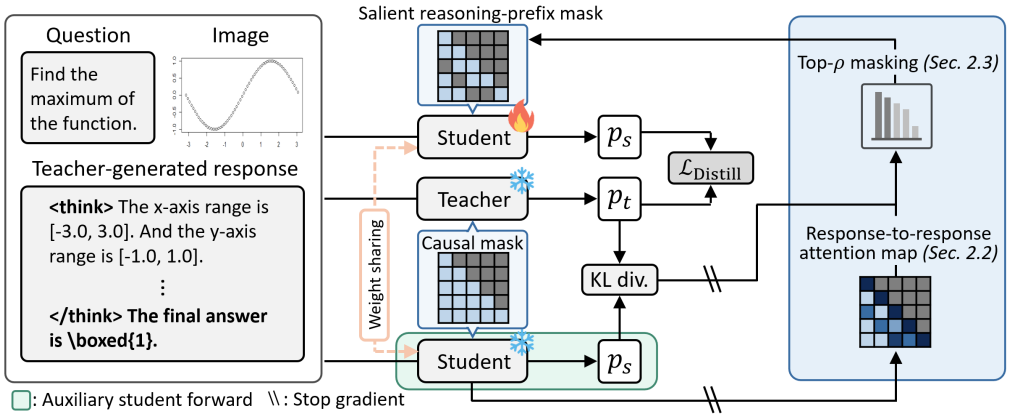


Figure 3: The illustration of Masking-KD. During distillation, the student is guided by our salient reasoning-prefix mask that blocks access to both future tokens and salient reasoning prefixes, whereas the teacher operates under the causal mask. This salient reasoning-prefix mask is derived from two quantities extracted from auxiliary student forward under the causal mask: 1) a response-to-response attention map for identifying salient reasoning-prefix (Sec. 2.2), and 2) token-wise reverse KL divergence for adaptively deciding the masking strength for each token (Sec. 2.3).

- Extensive experiments validate the effectiveness of the proposed framework by surpassing recent open-source VLMs, VLM distillations, and self-distillations on multimodal reasoning benchmarks. Further analysis shows that it improves the student's visual-anchored thinking from diverse aspects.

2 Think-Answer Reasoning Distillation Framework

In this section, we present Masking-KD, a simple yet effective think-answer distillation that masks the accumulated salient reasoning prefixes of the student to guide its thinking process based on visual evidence. Unlike existing VLM distillation methods [14, 4, 6], which directly transfer the teacher's visual knowledge through visual attention maps or visual token relations, Masking-KD encourages the student to develop its own visual-anchored thinking via salient reasoning-prefix masking. By mimicking teacher-like distributions under missing textual cues, the student learns to rely more on visual evidence, providing valuable learning signals beyond conventional teacher–student alignment.

We begin with an overview of Masking-KD (Sec. 2.1), explaining how the student is distilled under masked reasoning cues. We then describe how the salient reasoning-prefix mask is constructed: 1) *token-wise salient reasoning-prefix masking* (Sec. 2.2), determining which reasoning prefixes are masked, and 2) *self-paced masking budget scheduling* (Sec. 2.3), deciding how many reasoning prefixes are masked. Both are defined adaptively for each next-token prediction and are implemented via an attention mask under the standard causal masking [28] used for auto-regressive language modeling, supporting token-wise different masking in a single forward pass.

2.1 Overview of Masking-KD

Our knowledge distillation framework (Fig. 3) is built upon reverse KL divergence [1] to align the student's predictive distribution p_s over the vocabulary $\mathcal{V}$ with the teacher's one p_t along the distilled token sequence $\mathbf{y} = \{y_1, \ldots, y_N\}$ of length N, given the input image $\mathbf{x}_v$ and question $\mathbf{x}_q$. To impose masked reasoning prefixes on the student, we modify the causal mask used in the auto-regressive language modeling so that, at each decoding step, the student is restricted from access to both future tokens and salient reasoning-context tokens, as follows:

$$\mathcal{L}_{\text{Distill}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{y\in\mathcal{V}} p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \tilde{\mathbf{M}}) \log \frac{p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \tilde{\mathbf{M}})}{p_t(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \mathbf{M})}. \tag{1}$$

Here, p_s and p_t are scaled by the distillation temperature τ and $\mathbf{y}_{<n}$ indicates the prefixed reasoning-context tokens $\{y_1, \ldots, y_{n-1}\}$ up to step n. $\mathbf{M}$ is the standard causal mask used for the teacher, while $\tilde{\mathbf{M}}$ indicates the salient reasoning-prefix mask for the student, which extends the causal mask $\mathbf{M}$ by additionally masking salient reasoning-prefix tokens, enforcing the student to infer each subsequent token with missing prefixes during distillation.

Auxiliary Student Forward. To construct the salient reasoning-prefix mask $\tilde{\mathbf{M}}$, we perform an auxiliary forward pass of the student θ_s under the vanilla causal mask $\mathbf{M}$, *i.e.*, $\theta_s(\mathbf{x}_v, \mathbf{x}_q, \mathbf{y}, \mathbf{m})$, where we extract two quantities for mask construction: (1) **response-to-response attention map** $\mathbf{A}^{\text{resp}}$, used to identify which prefixes to mask (Sec. 2.2), and (2) **token-wise reverse KL divergence** $\mathbf{r} = \{r_n\}_{n=1}^{N}$,, used to determine how many prefixes to mask (Sec. 2.3). Specifically, we compute these two quantities as follows.

(1) response-to-response attention map $\mathbf{A}^{\text{resp}}$: To obtain this, we first average attention maps across all H transformer layers and then restrict the averaged attention map to the response-token block:

$$\mathbf{A}^{\text{resp}} \leftarrow \mathbf{A}\restriction_{\mathcal{I}^{\text{resp}} \times \mathcal{I}^{\text{resp}}} \in \mathbb{R}^{N \times N}, \quad \text{where } \mathbf{A} = \frac{1}{H}\sum_{h=1}^{H} \text{Attn}^h(\hat{\mathbf{x}}_v^{h-1}, \hat{\mathbf{x}}_q^{h-1}, \hat{\mathbf{y}}^{h-1}, \mathbf{M}), \tag{2}$$

$\text{Attn}^h(\cdot)$ denotes the attention operation at the h-th transformer layer, $\restriction_{\mathbf{I}\times\mathbf{J}}$ indicates the submatrix restriction to the rows indexed by $\mathbf{I}$ and columns indexed by $\mathbf{J}$, and $\mathcal{I}^{\text{resp}}$ denotes the index set of response-token positions. In particular, $\hat{\mathbf{x}}_v^{h-1}$, $\hat{\mathbf{x}}_q^{h-1}$, and $\hat{\mathbf{y}}^{h-1}$ denote the visual, question, and response representations at the input of the h-th transformer layer, respectively. Under the causal mask $\mathbf{M}$, the first $n-1$ entries of n-th row from $\mathbf{A}^{\text{resp}}$ represent the attention values over prefixed textual tokens $\mathbf{y}_{<n} = \{y_1, \ldots, y_{n-1}\}$ used to predict y_n. We leverage these attention weights to identify the salient reasoning-prefix tokens in Sec. 2.2.

(2) token-wise reverse KL divergence $\mathbf{r}$: Under the standard causal mask $\mathbf{M}$, we calculate the reverse KL divergence between the student distribution p_s and the teacher distribution p_t over the vocabulary $\mathcal{V}$ at every response position, where p_t is reused from Eq. (1). Formally, this is given by:

$$\mathbf{r} = \{r_n\}_{n=1}^{N}, \quad \text{where } r_n = \sum_{y \in \mathcal{V}} p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \mathbf{M}) \log \frac{p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \mathbf{M})}{p_t(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \mathbf{M})}. \tag{3}$$

Here, $\mathbf{r}$ captures the distributional discrepancy between the teacher and the student at each token position over N response tokens, reflecting the distillation difficulty of every token in the distilled response. We use this quantity to determine the masking strength for each token in Sec. 2.3.

2.2 Token-wise Salient reasoning-prefix Masking

Throughout the distilled thought-answer trace, the student tends to rely heavily on a small subset of salient reasoning cues with disproportionately high attention, as illustrated in Fig. 2. By blocking access to such salient prefixes for the student, we promote the use of visual information to compensate for the masked salient reasoning cues. To this end, we propose *token-wise salient reasoning-prefix masking*, which selectively masks salient prefixes for each next-token prediction to reflect that the most influential contextual tokens differ across decoding steps.

Construct Salient Reasoning-prefix Mask. To create a salient reasoning-prefix mask $\tilde{\mathbf{M}}$ used for distillation in Eq. (1), we utilize the response-to-response attention map $\mathbf{A}^{\text{resp}}$ extracted from an auxiliary student forward (Sec. 2.1). At each decoding step n, the row of this map $\mathbf{A}^{\text{resp}}_{n,<n}$ captures how strongly the preceding textual tokens $\mathbf{y}_{<n} = \{y_1, \ldots, y_{n-1}\}$ contribute to predict the current token y_n. Based on this prefix attention, we collect salient tokens using a nucleus top-p [11] style rule, which we refer to as top-ρ masking, greedily selecting the highest-attended prefixes until their cumulative attention ratio reaches ρ_n. The collected prefixes form a salient prefix set $\mathcal{S}_n$ such that:

$$\sum_{j \in \mathcal{S}_n} \bar{\mathbf{A}}^{\text{resp}}_{n,j} \geq \rho_n, \quad \text{where } \bar{\mathbf{A}}^{\text{resp}}_{n,j} = \frac{\mathbf{A}^{\text{resp}}_{n,j}}{\sum_{k=1}^{n-1} \mathbf{A}^{\text{resp}}_{n,k}}. \tag{4}$$

Here, $\bar{\mathbf{A}}^{\text{resp}}_{n,j}$ denotes the attention score assigned to the j-th prefix token, normalized over all $n-1$ prefixes. The threshold ρ_n is a self-paced cumulative ratio for step n (introduced in Sec. 2.3). Full details are provided in Appendix D.1. We then construct the salient reasoning-prefix mask $\tilde{\mathbf{M}} \in \{-\infty, 0\}^{N \times N}$ by extending the standard causal mask $\mathbf{M}$ to additionally suppress the salient prefix position in $\mathcal{S}_n$. Specifically, each entry of the salient reasoning-prefix mask $\tilde{\mathbf{M}}$ is given by:

$$\tilde{\mathbf{M}}_{n,j} = \begin{cases} -\infty, & \text{if } j > n \quad \text{(Causal masking)} \\ -\infty, & \text{if } j \in \mathcal{S}_n \quad \text{(Salient prefix masking)} \\ 0, & \text{otherwise} \end{cases}, \quad \forall n \in \{1, \ldots, N\}. \tag{5}$$

The resulting $\tilde{\mathbf{M}}$ is added to the attention logits as an attention mask, so that when predicting the current token y_n, the student is prevented from attending to future tokens and salient reasoning prefixes. For clarity, we omit visual and question token positions from both this formula and Fig 3.

2.3 Self-Paced Masking Budget Scheduling

As reasoning prefixes gradually accumulate, the student can imitate the teacher's subsequent tokens more easily, since prefixes themselves provide increasingly sufficient information (analyzed in Sec. 3.4). This suggests that applying a uniform masking budget to every response token is suboptimal. To address this, we introduce *self-paced masking budget scheduling*, which adaptively allocates masking budget (*i.e.*, the amount of salient reasoning-prefix to mask) according to the distillation difficulty of each token. As a result, easier tokens (*i.e.*, those with lower distillation loss) receive stronger masking to recover weakened distillation signals, while already harder tokens (*e.g.*, those with higher distillation loss) have greater access to reasoning prefixes.

Self-Paced Cumulative Ratio Threshold. In our framework, the masking amount is controlled by a self-paced cumulative ratio threshold ρ_n in Eq. (4). To determine this, we utilize token-wise reverse KL divergence $\mathbf{r} = \{r_n\}_{n=1}^{N}$ obtained from the auxiliary student forward in Eq. (3). Here, $\mathbf{r}$ captures the distillation difficulty along the distilled trace of length N. From $\mathbf{r}$, we decide ρ_n, as follows:

$$\rho_n = \rho_{\min} + (\rho_{\max} - \rho_{\min}) \cdot \sigma(\tilde{r}_n - \mu_{\tilde{r}}), \quad \text{where } \tilde{r}_n = -\log(r_n + \epsilon), \quad \mu_{\tilde{r}} = \frac{1}{N}\sum_{i=1}^{N} \tilde{r}_i, \quad (6)$$

$\rho_{\min}$ and $\rho_{\max}$ denote pre-defined lower and upper bounds of $\rho_n \in [\rho_{\min}, \rho_{\max}]$, which control the overall masking amount. $\sigma(\cdot)$ is the sigmoid function that maps the score to $[0, 1]$. We transform each r_n into a log-scaled score $\tilde{r}_n = -\log(r_n + \epsilon)$ to compress the dynamic range of $\{r_n\}_{n=1}^{N}$, which stabilizes the resulting threshold ρ_n. The negative sign in $\tilde{r}_n = -\log(r_n + \epsilon)$ reverses the ordering of the scores, so that tokens with smaller reverse KL values receive larger ρ_n. We subtract the mean score $\mu_{\tilde{r}}$ before applying σ to center $\{\rho_n\}_{n=1}^{N}$ around the average of $\mathbf{r}$. As a result, tokens with average difficulty are mapped near the midpoint of $[\rho_{\min}, \rho_{\max}]$, while relatively harder and easier tokens are pushed toward $\rho_{\min}$ and $\rho_{\max}$, respectively.

3 Experiments

3.1 Experimental Setup

Dataset and Metric. For VLM distillation, we construct the distilled data from ViRK39K [29] dataset by extracting the teacher's think-answer traces using greedy decoding with a maximum length of 4096 tokens, using the instructions in Appendix D.3. For self-distillation, the student uses its own pre-extracted think-answer traces. In all cases, we keep only correct responses, yielding 19k, 15k, and 10k samples for the 8B, 4B, and 2B Qwen3-VL-Thinking [2], respectively. For evaluation, we report pass@1 results with a maximum generation of 4096 tokens on: 1) math and geometric reasoning: Geometry-3K [18], MathVista [17], We-Math [23], MMK12 [21], MathVerse [38]; 2) logical reasoning: LogicVista [34]; and 3) multi-discipline multimodal reasoning: MMMU-Pro [37].

Implementation Details. We build our framework on Qwen3-VL-Thinking [2]. The student is trained for 2 epochs using a learning rate of 1×10^{-6}, with a batch size of 1 and gradient accumulation over 512 steps. The auxiliary student forward (Sec. 2.1) uses a weight-shared student rather than a separately initialized one. The self-paced cumulative ratio ρ_n (Sec. 2.3) is bounded by $\rho_{\min} = 0.3$ and $\rho_{\max} = 0.5$. To stabilize training and prevent loss explosion, we exclude the prefix token directly preceding the current token from masking. We set the distillation temperature $\tau = 2$ in the reverse KL divergence of Eq. (1). All experiments are conducted on NVIDIA A100 80 GB GPUs: two GPUs for the 2B and 4B students, and four GPUs for the 8B student in the self-distillation.

3.2 Main Results

Comparison with Open-source VLMs . We compare Pass@1 results of our Masking KD with open-source VLM models [20, 41, 31, 35, 3] on multimodal reasoning benchmarks in Tab. 1. Masking-KD-8B is obtained by self-distilling Qwen3-VL-8B-Thinking using its own predictions, while Masking-KD-4B and -2B are distilled from the Qwen3-VL-8B-Thinking teacher. Because all compared VLMs report their results with different generation lengths, we re-evaluate them using greedy decoding with a maximum length of 4096 tokens to ensure a direct comparison. Our

Table 1: **Comparison with open-source VLMs.** Our Masking-KD employs Qwen3-VL-Thinking models. † denotes the self-distilled 8B model using its own self-teacher, and ‡ indicates the distilled student from the 8B teacher. We evaluate all compared VLMs using greedy decoding with a maximum length of 4096 for direct comparison. Results on other VLM models are provided in Appendix A.1.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
∼8B Models								
Qwen3-VL-8B-Thinking	54.58	65.20	66.15	42.55	63.81	43.40	39.83	53.65
Ovis2-8B [20]	42.43	68.20	64.66	48.15	61.19	43.62	16.18	49.20
InternVL3.5-8B [31]	44.59	68.50	56.61	44.95	53.26	37.81	38.50	49.17
InternVL3-8B [41]	38.44	53.20	51.32	39.80	54.82	47.43	36.01	45.86
MiMo-VL-8B [35]	62.23	69.50	42.41	44.10	18.12	46.76	34.57	45.38
Qwen2.5-VL-7B [3]	40.43	67.50	48.74	43.90	38.67	46.31	31.56	45.30
Self-distill Masking-KD-8B† (ours)	**58.24**	**67.10**	**71.72**	**49.95**	**67.84**	**48.10**	**43.47**	**58.06**
∼4B Models								
Qwen3-VL-4B-Thinking	43.93	62.60	49.37	31.55	49.86	39.37	32.08	44.11
Ovis2-4B [20]	37.77	61.10	60.29	39.10	58.03	39.60	12.60	44.07
InternVL3.5-4B [31]	41.93	52.10	45.46	25.80	43.21	27.29	29.77	37.94
Qwen2.5-VL-3B [3]	26.29	55.90	49.66	39.85	40.69	38.03	27.75	39.74
Masking-KD-4B‡ (ours)	**52.58**	**66.50**	**71.03**	**51.00**	**62.66**	**52.35**	**40.52**	**56.66**
∼2B Models								
Qwen3-VL-2B-Thinking	26.29	43.10	25.17	13.00	28.21	18.57	14.51	24.12
Ovis2-2B [20]	31.11	54.70	51.95	32.45	50.32	31.77	10.23	37.50
InternVL3.5-2B [31]	29.95	40.60	24.31	14.70	27.94	16.11	12.54	23.74
InternVL3-2B [41]	33.78	46.70	47.93	37.00	40.37	33.33	22.95	37.44
Masking-KD-2B‡ (ours)	**40.93**	**59.20**	**63.79**	**37.20**	**57.89**	**41.61**	**30.75**	**47.34**

Table 2: **Result on self-distillation.** The base model is distilled using its own predictions under each method. The details on self-distillation are elaborated in Appendix D.2.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Base *Qwen3-VL-8B-Thinking*	54.58	65.20	66.15	42.55	63.81	43.40	39.83	53.65
w/ OPSD [39]	56.42	66.20	67.37	44.62	64.67	45.29	41.33	55.13
w/ Masking-KD (ours)	**58.24**	**67.10**	**71.72**	**49.95**	**67.84**	**48.10**	**43.47**	**58.06**
Base *Qwen3-VL-4B-Thinking*	43.93	62.60	49.37	31.55	49.86	39.37	32.08	44.11
w/ OPSD [39]	48.75	60.80	58.74	35.10	56.51	39.82	32.83	47.51
w/ Masking-KD (ours)	**52.25**	**66.10**	**68.79**	**50.85**	**64.08**	**50.78**	**39.25**	**56.01**
Base *Qwen3-VL-2B-Thinking*	26.29	43.10	25.17	13.00	28.21	18.57	14.51	24.12
w/ OPSD [39]	26.46	44.20	26.09	14.00	31.79	19.24	15.84	25.37
w/ Masking-KD (ours)	**33.61**	**52.00**	**40.40**	**19.25**	**40.09**	**25.28**	**20.00**	**32.95**

Masking-KD achieves state-of-the-art performance across all model sizes. Notably, our compact 2B model outperforms the undistilled 4B model, and our 4B model surpasses the undistilled 8B model, indicating the effectiveness of ours. Results on other VLM models are provided in Appendix A.1.

Results on Self-Distillation. To validate the effectiveness of our masking approach in self-distillation settings, Tab. 2 reports Pass@1 performance of our methods compared with another self-distillation method [39]. In this experiment, each model is distilled using its own pre-extracted think-answer trajectories. While OPSD [39] uses the student's thought-answer trace as input to construct a self-teacher signal, our method instead performs self-distillation with masked reasoning prefixes, forcing the model to recover its own predictions without relying on salient thinking cues. The superior performance of our methods over OPSD confirms the effectiveness of salient prefix masking in self-distillation. Further details on the self-distillation are provided in Appendix D.2.

Comparison with other VLM Distillations. To demonstrate the superiority of our Masking-KD, we compare its Pass@1 results with recent VLM distillation methods [4, 14, 6] on multimodal reasoning benchmarks. Tab. 3 and Tab. 4 present results for the 8B teacher – 4B student and 8B teacher – 2B student configurations, respectively. The naïve response distillation refers to the standard response-level distillation using reverse KL divergence. Since all compared methods are designed for distilling instruction-following abilities from scratch, we reproduce them in our think-answer distillation settings. In addition, these methods improve the student's visual perception ability by transferring the teacher's visual patterns, such as visual attention maps in CompoDistill [14], vision token relations in LLaVA-KD [4], and instruction-aware visual focus in Align-TI [6]. Despite

Table 3: **Comparison with other VLM distillations (8B teacher – 4B student).** We employ Qwen3-VL-Thinking for both teacher and student models. † denotes the methods proposed for distilling instruction-following ability; for this experiment, we reproduce them in our reasoning distillation setting. The results when combining our approach with them are reported in Appendix A.2.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Teacher *Qwen3-VL-8B-Thinking*	54.58	65.20	66.15	42.55	63.81	43.40	39.83	53.65
Student *Qwen3-VL-4B-Thinking*	43.93	62.60	49.37	31.55	49.86	39.37	32.08	44.11
Naïve Response Distillation	47.42	62.80	60.00	37.85	56.97	42.95	34.22	48.89
LLaVA-KD† [4]	49.75	63.80	62.59	39.45	59.27	43.40	34.28	50.36
CompoDistill† [14]	50.92	63.40	64.54	39.30	60.18	45.19	35.49	51.29
Align-TI† [6]	50.58	62.80	62.75	39.15	58.30	42.95	35.38	50.27
Masking-KD (ours)	**52.58**	**66.50**	**71.03**	**51.00**	**62.66**	**52.35**	**40.52**	**56.66**

Table 4: **Comparison with other VLM distillations (8B teacher – 2B student).**

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Teacher *Qwen3-VL-8B-Thinking*	54.58	65.20	66.15	42.55	63.81	43.40	39.83	53.65
Student *Qwen3-VL-2B-Thinking*	26.29	43.10	25.17	13.00	28.21	18.57	14.51	24.12
Naïve Response Distillation	35.94	54.50	51.38	26.10	48.67	28.64	22.60	38.26
LLaVA-KD† [4]	38.27	55.30	56.32	26.45	51.10	30.87	24.05	40.34
CompoDistill† [14]	38.94	57.50	57.07	28.30	49.50	34.80	24.51	41.52
Align-TI† [6]	38.27	56.60	53.97	27.95	49.36	33.33	24.05	40.50
Masking-KD (ours)	**40.93**	**59.20**	**63.79**	**37.20**	**57.89**	**41.61**	**30.75**	**47.34**
Undistilled *Qwen3-VL-4B-Thinking*	43.93	62.60	49.37	31.55	49.86	39.37	32.08	44.11

Table 5: **Extension to supervised fine-tuning (SFT).** The 2B student is trained on the 8B teacher-generated responses with one-hot labels (*i.e.*, without the teacher's logits).

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
SFT	35.77	52.60	48.39	23.85	45.00	26.85	19.88	36.05
SFT + ours	**40.60**	**55.00**	**57.99**	**33.40**	**54.50**	**34.45**	**26.01**	**43.14**

not explicitly distilling such visual patterns, Masking-KD achieves the best results, showing that reasoning-prefix masking is more effective in VLM reasoning distillation. Results obtained by combining our method with other VLM distillation approaches are reported in Appendix A.2.

Extension to Supervised Fine-Tuning. Since salient reasoning-prefix masking does not inherently depend on the distillation scenario, we further apply it to standard supervised fine-tuning (SFT), which is useful when only teacher-generated responses are available (*e.g.*, closed-source or API-only teachers). In this setting, the 2B student is trained on the 8B teacher-generated responses with one-hot labels, and the token-wise reverse KL divergence in self-paced masking budget scheduling is replaced with the per-token cross-entropy loss. As shown in Tab. 5, our masking consistently improves SFT across all benchmarks, demonstrating its generality beyond distillation.

Robustness to Perturbed Reasoning Prefixes. To verify that salient reasoning-prefix masking reduces the student's dependence on exposed reasoning prefixes, we intentionally perturb the reasoning prefixes at inference and compare the resulting performance degradation with naïve distillation. Specifically, on 400 problems, we perturb the student's reasoning prefixes at every decoding step in two ways: replacing numerical values with incorrect ones (*corrupted prefixes*; Tab. 6a) and randomly removing prefix tokens (*dropped prefixes*; Tab. 6b). As shown in Tab. 6, Masking-KD shows much smaller degradation than naïve distillation under both perturbations, indicating its reduced dependence on reasoning prefixes.

3.3 Ablation Study

We conduct ablation studies using the 8B teacher and 2B student from Qwen3-VL-Thinking models.

Effects of Each Proposed Method. In Tab. 7, we analyze the contribution of each method in our framework. Starting with the naïve response distillation, where the student is distilled from teacher-generated responses using reverse KL divergence and the causal mask. Guiding the student with our salient reasoning-prefix mask (Sec. 2.2) with a static cumulative ratio threshold, yields significant performance improvements. This result demonstrates that masking salient prefixes leads to more effective reasoning distillation. Introducing self-paced masking budget scheduling (Sec. 2.3) further

Table 6: **Robustness to perturbed reasoning prefixes.** The student's reasoning prefixes are perturbed at every decoding step at inference by (a) replacing numerical values with incorrect ones or (b) randomly removing prefix tokens.

(a) Corrupted prefixes

Corruption Ratio	Clean	30%	50%	100%
Naïve Distillation	40.50	37.25 (−8.0%)	36.50 (−9.9%)	33.50 (−17.3%)
Masking-KD (ours)	**53.00**	**52.50** (−0.9%)	**50.00** (−5.7%)	**47.75** (−9.9%)

(b) Dropped prefixes

Drop Ratio	Clean	10%	30%	50%
Naïve Distillation	40.50	39.00 (−3.7%)	38.50 (−4.9%)	34.25 (−15.4%)
Masking-KD (ours)	**53.00**	**53.00** (0.0%)	**51.75** (−2.4%)	**48.50** (−8.5%)

Table 7: **Ablation on each proposed method.** We begin with the naïve response distillation. Extensive ablation studies are provided in Appendix B.

Token-wise Salient Reasoning-prefix Masking	Self-Paced Masking Budget Scheduling	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
		35.94	54.50	51.38	26.10	48.67	28.64	22.60	38.26
✓		40.03	57.50	62.46	35.70	57.75	38.70	29.23	45.91
✓	✓	**40.93**	**59.20**	**63.79**	**37.20**	**57.89**	**41.61**	**30.75**	**47.34**

enhances performance by adaptively controlling the masking scale for each distilled token based on its distillation difficulty. This suggests that easily imitated tokens benefit from stronger masking, as they otherwise provide weak learning signals, while more difficult tokens require greater access to thought context for stable distillation. Extensive ablation studies are provided in Appendix B.

Ablation on Masked Region. Our methods apply the masking to prefixed response tokens. In this ablation (Tab. 8), we study different masking regions, such as visual tokens or question tokens, to validate the effectiveness of our response-prefix masking. For visual and question masking, we extract response-to-visual or response-to-question attention from the auxiliary student forward, instead of response-to-response attention, and apply the same top-ρ masking (Eq. (4)) within the selected region. All other components (*e.g.*, self-paced masking budget scheduling and hyperparameters) remain unchanged. The response-prefix masking achieves the best results, whereas the masking of visual or question tokens performs worse than naïve response distillation. In contrast, our reasoning-prefix masking suppresses exposed textual cues, encouraging the student to draw on other available information. Further analysis on these matters is discussed in Sec. 3.4.

Table 8: Masked regions.

Masked Region	Avg.
Visual tokens	37.39
Question tokens	31.31
Response (ours)	**47.34**
Naïve Distillation	38.26

3.4 Effect of Salient Reasoning-prefix Masking

Visual-anchored Thinking. To evaluate whether our method improves the student's use of visual information during reasoning, we compare the visual attention ratio over generation with the undistilled student (*i.e.*, Qwen3-VL-2B-Thinking [2]) and CompoDistill [14] in Fig. 4a. We also visualize the attention maps at the peak attention point in Fig. 4b. For robust analysis, we average the results over 10K responses generated by each model. The undistilled student shows a rapid decline in visual attention, indicating visual forgetting [27], while CompoDistill only partially mitigates this degradation. In contrast, our method maintains the highest visual attention throughout generation, showing its effectiveness in alleviating visual forgetting and promoting visual-anchored thinking.

Comparison on Visual Attention Map. In Fig. 5, we compare the visual attention maps over the entire thinking trace with the undistilled student, naïve KD, and CompoDistill [14] to highlight our superiority in enhancing the visual perception ability of the student. To obtain these visual attention maps, we average the visual attention scores over the entire thinking trace. Although CompoDistill directly distills the teacher's visual attention maps, Masking-KD attends more strongly to relevant image regions. This shows that our method more effectively encourages the student to exploit visual evidence during reasoning. More visualizations are provided in Appendix C.5.

Exploiting Visual Evidence during Distillation. In Fig. 6, we qualitatively compare the prediction behavior of the student with and without our salient reasoning-prefix masking. The figure visualizes which response prefixes are masked at a given decoding step and how the visual attention map is activated when predicting the current token. Without salient masking, the student relies on salient textual prefixes in the response, resulting in relatively weak attention to the image. In contrast, our masking strategy removes these highly influential prefixes, leading the student to compensate

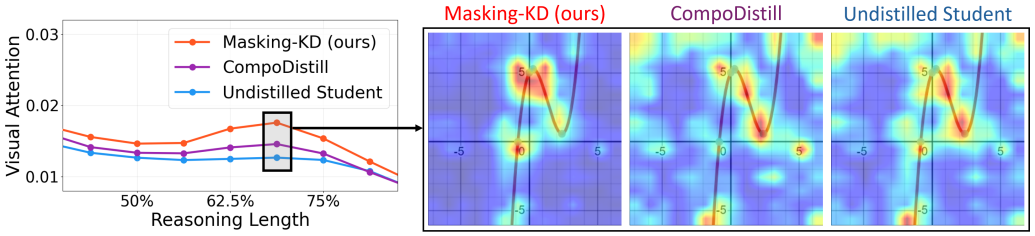


Figure 4: **Evidence on visual-anchored thinking.** (a) changes in visual attention as generation proceeds, and (b) an example of visual attention maps at the peak attention point in gray box.

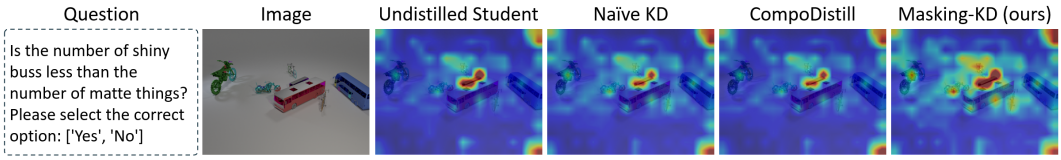


Figure 5: **Comparison on visual attention map.** We average the visual attention scores over the entire thinking trace. More visualizations are present in Fig. 10 of Appendix C.5.

for them by exploiting visual evidence. This leads to stronger activation in relevant image regions during distillation, encouraging that salient reasoning-prefix masking improves the student's visual perception throughout its thinking process. More visualizations are provided in Appendix C.6.

Evidence on Textual Shortcut Learning. In Fig. 7a, we analyze the reverse KL divergence along the response, averaged over 19K teacher-generated responses, to examine whether the student relies on exposed reasoning prefixes during distillation. Under naïve distillation, the reverse KL divergence gradually decreases as reasoning prefixes accumulate, indicating that the student can easily imitate subsequent tokens from exposed textual cues, whereas it increases under Masking-KD as these cues are masked. We further compare the training loss when masking different regions in Fig. 7b, where each region is masked with the same procedure as in Tab. 8. Masking visual tokens barely changes the loss from naïve distillation, while masking response prefixes yields the highest loss, showing that the student relies heavily on response prefixes when imitating the teacher. Note that the reverse KL divergence in Fig. 7a is measured without the distillation temperature, whereas the training loss in Fig. 7b is computed with the distillation temperature τ during distillation, resulting in their different scales.

Takeaway: our key findings

- When distilling long traces from think-answer VLMs, gradually accumulated reasoning prefixes offer sufficient information, allowing the student to easily imitate the teacher through exposed textual cues, reducing the need to refer to visual evidence (Fig. 6 and Fig. 7).
- Masking salient reasoning prefixes during distillation encourages the student to exploit more visual information to compensate for missing textual cues (Fig. 6), improving the student's visual-anchored thinking (Fig. 4 and Fig. 5).
- Rather than explicitly transferring the teacher's visual knowledge, guiding the student itself to anchor its own reasoning in visual information is effective for VLM reasoning distillation. (Tab. 3 and Tab. 4).

4 Related Work

Think-answer Reasoning. Recent think-answer models, such as DeepSeek-R1 [8] and OpenAI o1 [13], have demonstrated that explicitly generating intermediate reasoning before the final answer can substantially improve performance on complex reasoning tasks [10, 9]. This think-answer paradigm has recently been extended to vision-language models (VLMs) to enable longer and more deliberate reasoning trajectories for multimodal problem solving. For example, VL-Rethinker [29] strengthens slow-thinking behavior through reinforcement learning [24] and explicit rethinking. R1-OneVision [36] bridges visual perception and deep reasoning through cross-modal formalization and

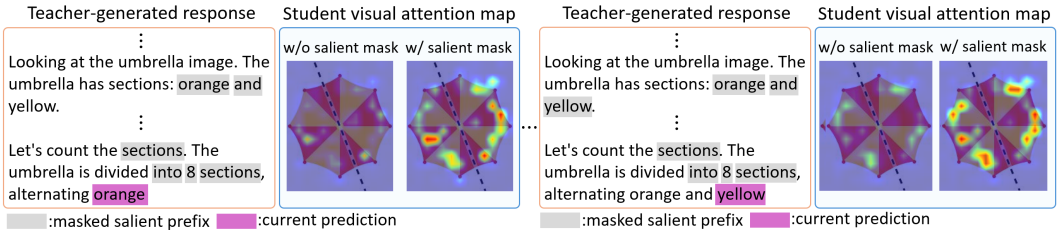
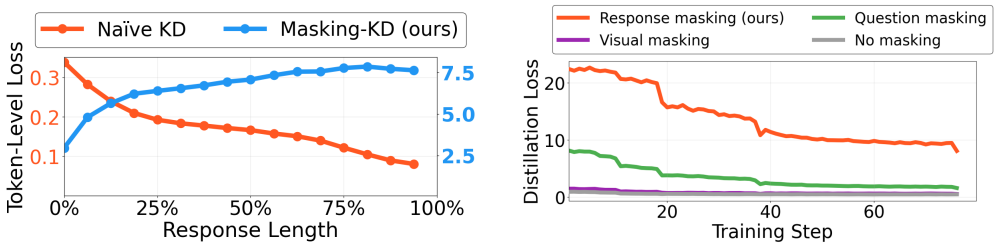


Figure 6: **Prediction behavior of the student during distillation** without and with our salient reasoning-prefix mask. Without a salient mask, the student uses a standard causal mask to predict the current token. With a salient mask, the student exploits more visual information to compensate for the masked salient reasoning prefix. More visualizations are provided in Fig. 12 of Appendix C.6.



(a) Decay of Reverse KL Divergence.

(b) Distillation Loss Comparison.

Figure 7: **Evidence on textual shortcut learning.** (a) Reverse KL divergence along the response. (b) Training loss when masking different regions as in Tab. 8, whose scale differs from (a) due to the distillation temperature τ.

step-wise reasoning supervision. However, larger think-answer VLMs introduce high computational costs, motivating knowledge distillation into compact VLMs, which we explore in this work.

Knowledge Distillation in VLMs. Knowledge distillation (KD) is widely used to compress vision-language models (VLMs) by transferring capabilities from a larger teacher to a smaller student. Existing VLM distillation methods [6, 25, 5, 4, 14, 7] typically focus on preserving the teacher's visual knowledge through intermediate supervision, such as visual relation distillation in LLaVA-KD [4], visual attention alignment in CompoDistill [14], and instruction-aware visual focus in Align-TI [6]. While effective for conventional VLM distillation, these methods do not address a challenge unique to think-answer VLMs: as long reasoning traces unfold, accumulated reasoning prefixes become highly informative, allowing the student to imitate the teacher through exposed textual cues rather than sustaining visually anchored reasoning. To address this, we propose a first think-answer distillation framework that suppresses shortcut textual cues, encouraging the student to rely on the remaining multimodal evidence by masking prefixed salient reasoning cues.

Self-Distillation. Self-distillation has emerged as an effective paradigm for improving model performance without a separate, larger teacher. In LLMs, OPSD [39] uses the student's own reasoning trace as input to a self-teacher, which then provides feedback to improve the student. In VLMs, SDRT [33] self-distills from diverse reasoning traces generated by the model itself, training the student to imitate multiple reasoning paths. While effective, these methods often require extra prompt design and multi-trace construction to create self-feedback signals. In contrast, our approach simply masks salient reasoning prefixes for the student, while the self-teacher observes the full reasoning context, encouraging the student to exploit visual evidence to recover the missing textual cues.

5 Conclusion

In this paper, we present Masking-KD, a novel think-answer VLM distillation framework that masks salient reasoning prefixes to prevent the student from over-relying on exposed textual cues during distillation. By encouraging the student to exploit visual evidence for recovering missing reasoning context, our method promotes more visual-anchored thinking during generation. Extensive experiments and analyses demonstrate that Masking-KD achieves outstanding results on multimodal reasoning benchmarks and effectively alleviates visual forgetting, arised in think-answer VLMs.

Acknowledgements

This work was supported by the AI Graduate School Program at POSTECH (RS-2019-II191906 (5%)), the IITP grants (RS-2026-25518317 (50%), RS-2026-25616370 (30%)), and the InnoCORE program (N10250156 (15%)) funded by MSIT,Korea.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024. 3, 16, 18

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 5, 8, 15, 16, 17, 18, 20

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 6

[4] Yuxuan Cai, Jiangning Zhang, Haoyang He, Xinwei He, Ao Tong, Zhenye Gan, Chengjie Wang, Zhucun Xue, Yong Liu, and Xiang Bai. Llava-kd: A framework of distilling multimodal large language models. In *ICCV*, 2025. 1, 3, 6, 7, 10, 15, 16

[5] Jiajun Cao, Yuan Zhang, Tao Huang, Ming Lu, Qizhe Zhang, Ruichuan An, Ningning Ma, and Shanghang Zhang. Move-kd: Knowledge distillation for vlms with mixture of visual encoders. In *CVPR*, 2025. 10

[6] Lin Chen, Xiaoke Zhao, Kun Ding, Weiwei Feng, Changtao Miao, Zili Wang, Wenxuan Guo, Ying Wang, Kaiyuan Zheng, Bo Zhang, et al. Beyond next-token alignment: Distilling multimodal large language models via token interactions. *arXiv preprint arXiv:2602.09483*, 2026. 1, 3, 6, 7, 10, 15, 16

[7] Qianhan Feng, Wenshuo Li, Tong Lin, and Xinghao Chen. Align-kd: Distilling cross-modal alignment knowledge for mobile vision-language model. *CVPR*, 2025. 1, 10

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1, 9

[9] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021. 9

[10] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS Datasets and Benchmarks*, 2021. 9

[11] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020. 4, 22

[12] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength natural language processing in Python. Zenodo, 2020. 20, 21

[13] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 9

[14] Jiwan Kim, Kibum Kim, Sangwoo Seo, and Chanyoung Park. Compodistill: Attention distillation for compositional reasoning in multimodal llms. *ICLR*, 2026. 1, 3, 6, 7, 8, 10, 15, 16

[15] Byung-Kwan Lee, Youngchae Chee, and Yong Man Ro. Recursive think-answer process for llms and vlms. *CVPR Findings*, 2026. 1

[16] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023. 17

[17] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024. 1, 5

[18] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Intergps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *ACL*, 2021. 5

[19] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. 2022. 1

[20] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model, 2024. *arXiv preprint arXiv:2405.20797*, 2024. 5, 6

[21] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning, 2025. 5

[22] OpenAI. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026. 1

[23] Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *ACL*, 2025. 5

[24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 9

[25] Haoyi Sun, Xiaoxiao Wang, Ning Mao, Qian Wang, Lifu Mu, Wen Zheng, Tao Wei, and Wei Chen. Switch-kd: Visual-switch knowledge distillation for vision-language models. *CVPR Findings*, 2026. 1, 10

[26] The Gemini Team. Gemini 3.1 pro: A smarter model for your most complex tasks, 2026. 1

[27] Xinyu Tian, Shu Zou, Zhaoyuan Yang, Mengqi He, Fabian Waschkowski, Lukas Wesemann, Peter Tu, and Jing Zhang. More thought, less accuracy? on the dual nature of reasoning in vision-language models. *ICLR*, 2026. 2, 8

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 3

[29] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. In *NeurIPS*, 2025. 1, 5, 9, 22

[30] Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. In *NeurIPS*, 2025. 2

[31] 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. 5, 6, 15

[32] Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, et al. Perception-aware policy optimization for multimodal reasoning. In *ICLR*, 2026. 1, 2

[33] Guande Wu, Huan Song, Yawei Wang, Qiaojing Yan, Yijun Tian, Lin Lee Cheong, and Panpan Xu. Sdrt: Enhance vision-language models by self-distillation with diverse reasoning traces. *arXiv preprint arXiv:2503.01754*, 2025. 10

[34] Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. LogicVista: Multimodal llm logical reasoning benchmark in visual contexts. In *ICLR*, 2025. 5

[35] LLM-Core-Team Xiaomi. Mimo-vl technical report. *arXiv preprint arXiv:2506.03569*, 2025. 5, 6

[36] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *ICCV*, 2025. 1, 9

[37] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *ACL*, 2025. 1, 5

[38] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *ECCV*, 2024. 1, 5

[39] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. 6, 10

[40] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025. 1

[41] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 5, 6

Hide to See: Reasoning-prefix Masking for Visual-anchored Thinking in VLM Distillation

- Appendix -

Overview of Appendix

We provide the table of contents for the Appendix below:

A. Additional Experiments
- A.1. Results on other VLM Models
- A.2. Compatibility with other VLM Distillations
- A.3. Results from Student-generated Response
- A.4. Computational Comparison with other VLM Distillations
- A.5. Statistical Significance
- A.6. Evaluation on Hallucination Benchmark

B. Additional Ablation Studies
- B.1. Ablation Study within Proposed Method
- B.2. Auxiliary Student Forward
- B.3. Loss Functions
- B.4. Excluding Immediate Previous Token from Masking

C. Additional Analyses
- C.1. Effectiveness on Longer Reasoning Traces
- C.2. Analysis on Self-Paced Masking Budget
- C.3. Statistics of Masked Prefix Positions
- C.4. Example of Inference
- C.5. More Comparison on Visual Attention Map
- C.6. More Prediction Behavior of the Student during Distillation

D. Additional Details
- D.1. Details on Top-ρ Masking
- D.2. Details on Self-Distillation
- D.3. Instruction for Teacher-generated Response

E. Further Discussion
- E.1. Limitation
- E.2. Social Impact

A Additional Experiments

A.1 Results on other VLM Models

In this section, we validate the generality of our approach by applying it to other VLM models across both knowledge distillation and self-distillation settings. We report results for InternVL3.5 [31] in Tab. 9 and Qwen3-VL-Instruct [2] in Tab. 10. Masking-KD achieves the best performance across various teacher-student configurations (*i.e.*, 8B–8B self-distillation, 8B–4B knowledge distillation, and 8B–2B knowledge distillation), demonstrating its generality across different VLM models.

Table 9: **Results on other VLM models (InternVL3.5 [31]).** † denotes a self-distilled model that uses its own predictions under our salient reasoning-prefix mask, and ‡ indicates the distilled student from the 8B teacher.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Teacher *InternVL3.5-8B*	44.59	68.50	56.61	44.95	53.26	37.81	38.50	49.17
Self-distill Masking-KD† (ours)	**46.26**	**69.70**	**58.79**	**45.45**	**54.97**	**39.16**	**39.84**	**50.60**
Student *InternVL3.5-4B*	41.93	52.10	45.46	25.80	43.21	27.29	29.77	37.94
Naïve Response Distillation	**44.93**	52.30	47.64	27.60	43.58	27.74	25.09	38.41
Masking-KD‡ (ours)	44.26	**56.40**	**50.75**	**30.45**	**47.61**	**30.20**	**27.23**	**40.99**
Student *InternVL3.5-2B*	29.95	40.60	24.31	14.70	27.94	16.11	12.54	23.74
Naïve Response Distillation	32.95	43.50	27.59	13.40	30.87	18.12	13.53	25.71
Masking-KD‡ (ours)	**35.61**	**52.00**	**41.95**	**19.70**	**38.67**	**25.73**	**20.46**	**33.45**

Table 10: **Results on other VLM models (Qwen3-VL-Instruct [2]).** † denotes a self-distilled model that uses its own predictions under our salient reasoning-prefix mask, and ‡ indicates the distilled student from the 8B teacher.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Teacher *Qwen3-VL-8B-Instruct*	54.58	67.40	70.11	58.20	63.49	54.36	38.90	58.15
Self-distill Masking-KD† (ours)	**59.23**	**67.70**	**74.48**	**65.00**	**68.35**	**55.26**	**44.80**	**62.12**
Student *Qwen3-VL-4B-Instruct*	51.91	65.10	58.28	46.10	45.05	45.64	19.83	47.42
Naïve Response Distillation	48.59	**64.90**	64.14	48.00	57.61	47.43	36.42	52.44
Masking-KD‡ (ours)	**54.74**	64.60	**65.98**	**57.70**	**61.06**	**51.45**	**40.12**	**56.52**
Student *Qwen3-VL-2B-Instruct*	26.46	55.70	33.56	25.60	30.00	27.07	20.46	31.26
Naïve Response Distillation	31.78	57.10	52.18	33.05	42.66	34.00	27.05	39.69
Masking-KD‡ (ours)	**36.77**	**58.00**	**55.52**	**44.15**	**48.67**	**43.18**	**30.29**	**45.23**

A.2 Compatibility with other VLM Distillations

In Tab. 11, we report the performance of Masking-KD when combined with other VLM distillation methods, including LLaVA-KD [4], CompoDistill [14], and Align-TI [6]. Our approach consistently improves performance across these methods. However, the performance gains are smaller when combined with naïve response distillation. We attribute this to a partial conflict between prior methods that explicitly distill the teacher's visual knowledge, and our objective for enhancing the student's own use of visual evidence. Nevertheless, the consistent improvements indicate our effectiveness in VLM distillations.

Table 11: **Compatibility with other VLM distillations.** We adapt our salient reasoning-prefix masking into other VLM distillation methods.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Naïve Response Distillation	35.94	54.50	51.38	26.10	48.67	28.64	22.60	38.26
w/ Masking-KD (ours)	40.93	59.20	63.79	37.20	57.89	41.61	30.75	47.34
LLaVA-KD [4]	38.27	55.30	56.32	26.45	51.10	30.87	24.05	40.34
w/ Masking-KD (ours)	43.09	58.40	64.43	36.20	57.75	37.14	29.31	46.62
CompoDistill [14]	38.94	57.50	57.07	28.30	49.50	34.80	24.51	41.52
w/ Masking-KD (ours)	42.60	58.90	60.63	32.35	55.96	33.46	27.86	44.54
Align-TI [6]	38.27	56.60	53.97	27.95	49.36	33.33	24.05	40.50
w/ Masking-KD (ours)	42.43	58.20	56.69	30.15	53.16	33.89	25.97	42.93

A.3 Results from Student-generated Response

In the main manuscript, we conduct distillation using teacher-generated responses. Here, we further show that our approach is also effective when distilling from student-generated responses, compared

with other VLM distillations [14, 6, 4] in Tab. 12 (8B teacher – 4B student) and Tab. 12 (8B teacher – 2B student). Using student-generated responses has the advantage of providing supervision that is closer to the student's own reasoning distribution, as described in [1]. This can reduce the distribution mismatch between training traces and the student's generation behavior, leading to more stable and better-aligned distillation. These results demonstrate that our approach remains effective even when distilling from student-generated responses.

Table 12: **Distillation Results from Student-generated Responses (8B teacher – 4B student).** For distillation, we use student-generated responses instead of teacher-generated responses, where the mismatched distribution between training traces and the student's generation behavior is alleviated.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Teacher *Qwen3-VL-8B-Thinking*	54.58	65.20	66.15	42.55	63.81	43.40	39.83	53.65
Student *Qwen3-VL-4B-Thinking*	43.93	62.60	49.37	31.55	49.86	39.37	32.08	44.11
Naïve Response Distillation	48.92	61.70	58.91	37.25	57.52	42.95	33.64	48.70
LLaVA-KD† [4]	49.58	63.50	60.86	36.95	57.89	42.51	33.64	49.28
CompoDistill† [14]	49.25	62.00	61.21	38.15	58.21	44.07	33.82	49.53
Align-TI† [6]	51.08	61.70	61.84	38.50	57.34	42.73	34.68	49.70
Masking-KD (ours)	**54.91**	**66.10**	**69.43**	**51.60**	**63.94**	**53.02**	**40.52**	**57.07**

Table 13: **Distillation Results from Student-generated Responses (8B teacher – 2B student).**

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Teacher *Qwen3-VL-8B-Thinking*	54.58	65.20	66.15	42.55	63.81	43.40	39.83	53.65
Student *Qwen3-VL-2B-Thinking*	26.29	43.10	25.17	13.00	28.21	18.57	14.51	24.12
Naïve Response Distillation	32.45	46.80	41.55	18.35	40.14	23.71	16.36	31.34
LLaVA-KD† [4]	33.11	50.60	44.05	18.85	42.32	21.58	17.75	32.61
CompoDistill† [14]	33.44	50.00	45.75	19.85	43.21	23.94	18.61	33.54
Align-TI† [6]	33.04	49.51	43.49	18.65	41.60	22.82	18.54	32.52
Masking-KD (ours)	**44.59**	**58.00**	**61.26**	**35.50**	**56.61**	**40.72**	**30.81**	**46.78**
Undistilled *Qwen3-VL-4B-Thinking*	43.93	62.60	49.37	31.55	49.86	39.37	32.08	44.11

A.4 Computational Comparison with other VLM Distillations

In Masking-KD, the auxiliary student forward pass increases computational overhead, and extracting attention maps incurs additional memory usage. We investigate these aspects compared with other VLM distillation methods, including na"ive response distillation, LLaVA-KD [4], CompoDistill [14], and Align-TI [6], in Tab. 14. For per-step time (s), we measure the time required to process 512 samples (one step in our training recipe) on two A100 GPUs, and report memory usage as the average over one step. The reported average Pass@1 results come from Tab. 4 of the main manuscript. We use the 8B teacher and the 2B student from Qwen3-VL-Thinking. Align-TI shows the highest computational overhead because it also requires extra forward passes and intermediate statistics. Other VLM distillation methods incur only marginal additional memory usage because they mainly handle partial attention maps, such as visual attention maps. In contrast, our method requires response-to-response attention maps over long reasoning traces, leading to relatively higher memory overhead. Despite the increased overhead, Masking-KD achieves superior performance.

Table 14: **Computational Comparison with other VLM distillations.** Per step (s) denotes the measured time required to process 512 samples (one step in our training recipe) on two A100 GPUs.

Method	Per step (s)	Memory (GB)	Avg. Pass@1
Naïve Response Distillation	75.8	36.5	38.3
LLaVA-KD [4]	100.4	37.2	40.3
CompoDistill [14]	103.9	38.0	41.5
Align-TI [6]	**381.2**	38.2	40.5
Masking-KD (ours)	157.9	**40.9**	**47.3**

A.5 Statistical Significance

To ensure the statistical significance of our results, we additionally run Masking-KD three times with different random seeds. Together with the original run reported in the main manuscript, we report the mean performance and standard deviation over four runs in total. For this experiment, we use the 8B teacher and 2B student from Qwen3-VL-Thinking [2]. As shown in Tab. 15, Masking-KD maintains low variance across runs. These results indicate that the performance gains of our method are stable and not sensitive to random seeds.

Table 15: **Statistical Significance.** We additionally run Masking-KD three times and report the mean performance with standard deviation denoted by ± over four runs in total.

Trials	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Run 1 (reported)	40.93	59.20	63.79	37.20	57.89	41.61	30.75	47.34
Run 2	43.93	58.60	63.56	38.05	57.98	41.39	30.81	47.76
Run 3	43.26	58.00	63.33	36.45	59.22	41.39	30.52	47.45
Run 4	41.76	58.70	63.28	36.85	57.34	42.06	30.40	47.20
Avg. ±std	42.47±1.16	58.63±0.50	63.49±0.20	37.14±0.71	58.11±0.71	41.61±0.27	30.62±0.18	47.44±0.23

A.6 Evaluation on Hallucination Benchmark

Since our method masks salient reasoning prefixes during distillation, one may be concerned that the masking could induce hallucination. To examine this, we evaluate our method in a zero-shot manner on POPE [16], a widely used benchmark for object hallucination in VLMs. For this experiment, we use the 8B teacher and 2B student from Qwen3-VL-Thinking [2]. As shown in Tab. 16, Masking-KD performs comparably to or slightly better than both the base model (*i.e.*, the undistilled Qwen3-VL-2B-Thinking) and naïve distillation, indicating that our masking does not induce hallucination.

Table 16: **Zero-shot evaluation on hallucination benchmark.** We report accuracy (Acc) and F1 score on the random, popular, and adversarial splits of POPE [16]. We use the 8B teacher and 2B student from Qwen3-VL-Thinking [2].

Method	Random		Popular		Adversarial		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Base *Qwen3-VL-2B-Thinking*	90.27	89.41	88.17	87.43	**87.17**	**86.60**	88.53	87.80
Naïve Distillation	90.83	90.13	88.50	87.91	86.87	86.41	88.73	88.12
Masking-KD (ours)	**91.17**	**90.57**	**88.53**	**88.08**	86.60	86.37	**88.77**	**88.31**

B Additional Ablation Studies

B.1 Ablation Study within Proposed Method.

Tab. 17 ablates our key design choices and hyperparameters within the proposed methods. For *token-wise salient reasoning prefix masking* (Tab. 17a), we observe: **(1) token-wise adaptive masking**, where adaptive masking outperforms a non-adaptive masking, which masks the same prefixes at every decoding step; and **(2) which reasoning prefixes to mask**, where masking high-attention prefixes achieves the best results over random, low-attention, and middle-attention masking. For *self-paced masking budget scheduling* (Tab. 17b), we study: **(3) threshold type**, where our cumulative ratio that masking prefixes according to their accumulated influence is more effective; and **(4) the range of cummulative ratio**, where $\rho_{\text{min}} = 0.3$ and $\rho_{\text{max}} = 0.5$ yield the best performance.

Table 17: **Ablation within each proposed method.**

(a) Token-wise salient reasoning-prefix masking

Ablations	Avg.
(1) Effect of token-wise adaptive masking	
Non-adaptive masking	41.73
Token-wise adaptive masking	**47.34**
(2) Which reasoning prefixes to mask	
Random prefixes	42.71
Low-attention prefixes	37.29
Middle-attention prefixes	38.84
High-attention (*i.e.*, salient) prefixes	**47.34**

(b) Self-paced masking budget scheduling

Ablations	Avg.
(3) Threshold type	
Attention threshold	46.29
Masking ratio %	46.21
Cumulative ratio ρ	**47.34**
(4) $[\rho_{\text{min}}, \rho_{\text{max}}]$ in cumulative ratio ρ	
[0.1,0.3]	45.76
[0.5,0.7]	45.82
[0.3,0.5]	**47.34**

B.2 Auxiliary Student Forward

To construct the salient reasoning-prefix mask, we use an auxiliary student forward pass (Sec. 2.1) to obtain the response-to-response attention map $\mathbf{A}^{\text{reps}}$ and token-wise reverse KL divergence $\mathbf{r}$. In Tab. 18, we compare our weight-shared auxiliary student with a separately initialized frozen student. The weight-shared design performs best, showing that adapting the mask construction to the current student state is effective.

Table 18: **Ablation within Auxiliary Student Forward**. † denotes using an auxiliary forward pass from a separately initialized frozen student, instead of the currently distilled student (reported).

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
No weight-sharing†	**43.93**	58.40	63.75	34.20	**58.85**	40.04	28.27	46.78
Weight-sharing (reported)	40.93	**59.20**	**63.79**	**37.20**	57.89	**41.61**	**30.75**	**47.34**

B.3 Loss Functions

Our distillation framework is built upon reverse KL divergence [1] between the student's predictive distribution and the teacher's distribution, as described in Sec. 2.1. However, forward KL divergence and mixed KL objectives, which combine forward and reverse KL divergence with equal weights, are also widely used in distillation. Here, we investigate the effect of these distillation losses in Tab. 19. Using reverse KL divergence alone achieves the best results, indicating that its mode-seeking behavior [1] is more effective for encouraging the student to focus on the teacher's high-probability predictions, rather than matching the entire distribution (*e.g.*, forward KL div.).

Table 19: **Ablation on Loss Function**. † denotes the mixed KL objective, which combines forward KL divergence and reverse KL divergence with equal weights of 0.5.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Forward KL Div.	42.60	57.00	59.14	35.35	54.27	38.70	28.38	45.06
Mixed KL Div.†	**42.61**	57.80	60.92	35.35	56.65	37.14	30.17	45.80
Reverse KL Div. (reported)	40.93	**59.20**	**63.79**	**37.20**	**57.89**	**41.61**	**30.75**	**47.34**

B.4 Excluding Immediate Previous Token from Masking

As described in Implementation Details of Sec. 3.1, we exclude the immediate previous prefix token from masking to stabilize training and prevent loss explosion. Here, we ablate this design in Tab. 20 by including the immediate previous prefix token in the masking candidates. When this token is included in masking, the student struggles to predict subsequent tokens, significantly degrading performance. Thus, excluding it from masking preserves essential local context for next-token prediction while maintaining training stability.

Table 20: **Ablation on excluding immediate previous token from masking.** As elaborated in Impplementation Details of Sec. 3.1 from the main manuscript, we exclude this from masking to stable training and prevent the loss explosion, since excluding the immediate previous token from masking preserves essential local context for next-token prediction.

Method	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
Including immediate previous token in masking	**41.43**	57.70	54.89	26.35	51.06	30.43	24.51	40.91
Excluding immediate previous token from masking (ours)	40.93	**59.20**	**63.79**	**37.20**	**57.89**	**41.61**	**30.75**	**47.34**

C Additional Analyses

C.1 Effectiveness on Longer Reasoning Traces

In this section, we investigate whether the advantage of Masking-KD becomes larger on longer reasoning traces by comparing traces of different total lengths through distillation with longer traces and length-binned analyses of performance and visual attention. Since textual shortcut learning arises from reasoning prefixes accumulated over a trace (Sec. 3.4), its effect, and thus the benefit of Masking-KD, is expected to grow with trace length. Note that the analyses in Fig. 4 and Fig. 7a of the main manuscript examine changes along relative positions within each trace, rather than across traces of different total lengths. For all experiments, we use the 8B teacher and 2B student from Qwen3-VL-Thinking [2].

Distillation with Longer Traces. We first study whether Masking-KD benefits further from longer reasoning traces. To this end, we construct the distillation data from teacher traces generated with a maximum length of 8,192 tokens, in addition to the default setting with a maximum length of 4,096 tokens (Sec. 3.1). This yields 24K correct traces, from which we randomly sample 19K traces to match the size of the default training set for a fair comparison. As shown in Tab. 21, Masking-KD

obtains a clear gain when trained on longer traces (47.34 → 51.01 in average Pass@1), indicating that the advantage of our approach is further amplified on longer reasoning traces.

Table 21: **Distillation with longer reasoning traces.** We construct the distillation data from teacher traces generated with a maximum length of 8,192 tokens, instead of 4,096 tokens (default), and randomly sample 19K correct traces to match the size of the default training set.

Max. Length	Geo3k	MathVista	We-Math	MMK12	MathVerse	LogicVista	MMMU^{Pro}	Avg.
4,096 (default)	40.93	**59.20**	63.79	37.20	57.89	41.61	30.75	47.34
8,192	**46.09**	58.50	**66.38**	**46.45**	**64.17**	**41.83**	**33.64**	**51.01**

Trace-Length Distribution. In Tab. 22, we report the trace-length distribution of the training data to clarify how trace lengths differ between the two settings and whether the default training set is affected by truncation. The default training set contains no truncated traces, since a trace that reaches the 4,096-token limit fails to produce a final answer and is therefore removed when we keep only correct responses (Sec. 3.1). Thus, all 19K training traces in the default setting contain complete final answers within the 4,096-token limit, while the 8,192-token setting provides substantially longer traces (*e.g.*, 4,013 vs. 7,818 tokens at p99).

Table 22: **Trace-length distribution of training data.** We report the percentiles of the number of tokens in the training traces for each setting. For the 8,192-token setting, the statistics are computed over the 19K traces used for training.

Max. Length	p1	p25	p50	p75	p99
4,096 (default)	221	951	1,552	2,492	4,013
8,192	236	1,096	1,978	3,648	7,818

Performance by Trace Length. We further analyze the relationship between trace length and performance gains by comparing the average Pass@1 of naïve distillation and Masking-KD across trace-length bins in Tab. 23. For a paired comparison, we assign each evaluation example to a bin according to the length of the response generated by the 8B teacher, so that both methods are evaluated on exactly the same examples within each bin. Across the first three bins, the gain of Masking-KD gradually increases as the trace length grows (+7.55, +20.43, and +27.77). The last bin (3,072+) is strongly affected by the 4,096-token evaluation limit, as many generations in this bin reach the maximum length without producing a final answer and are therefore counted as incorrect. We thus consider the result in this bin inconclusive under the current evaluation limit, rather than evidence against the length-dependent trend.

Visual Attention by Trace Length. In Tab. 23, we also compare the average visual attention of naïve distillation and Masking-KD using the same bins and evaluation examples, verifying whether the student's use of visual evidence also depends on trace length. While both methods show comparable visual attention in the shortest-trace bin, Masking-KD achieves consistently higher visual attention across all longer-trace bins (+12.9% to +17.2%), further supporting its advantage on longer reasoning traces.

Table 23: **Average Pass@1 and visual attention by trace length.** Each evaluation example is assigned to a bin according to the length of the response generated by the 8B teacher, so that naïve distillation and Masking-KD are evaluated on exactly the same examples within each bin. Gain denotes the difference between Masking-KD and naïve distillation, and relative gain denotes this difference normalized by naïve distillation.

Trace Length	0–1,024	1,024–2,048	2,048–3,072	3,072+
Average Pass@1				
Naïve Distillation	69.85	40.82	18.32	**8.89**
Masking-KD (ours)	**77.40**	**61.25**	**46.09**	8.43
Gain	+7.55	+20.43	+27.77	−0.46
Average Visual Attention				
Naïve Distillation	**0.0309**	0.0207	0.0157	0.0140
Masking-KD (ours)	0.0300	**0.0242**	**0.0184**	**0.0158**
Relative Gain	−2.9%	+16.9%	+17.2%	+12.9%

C.2 Analysis on Self-Paced Masking Budget

In this section, we analyze how our self-paced masking budget scheduling (Sec. 2.3) allocates the masking budget ρ_n and how it differs from a static budget. For comparison, we consider a static-budget setting with a fixed $\rho_n = 0.4$ for all tokens, which is the midpoint of our range $[\rho_{\text{min}}, \rho_{\text{max}}] = [0.3, 0.5]$. For all experiments, we use the 8B teacher and 2B student from Qwen3-VL-Thinking [2].

Distillation Signal over Training. We first compare the distillation signal of the two settings over training progress in Tab. 24, where the signal is measured by the masked-context reverse KL divergence (*i.e.*, $\mathcal{L}_{\text{Distill}}$ in Eq. (1)). In both settings, the signal naturally decays as optimization proceeds. However, self-paced scheduling consistently maintains a stronger signal throughout training, leading to higher average Pass@1 (45.91 $\rightarrow$ 47.34).

Table 24: **Distillation signal over training.** We report the masked-context reverse KL divergence (*i.e.*, $\mathcal{L}_{\text{Distill}}$ in Eq. (1)) averaged over each interval of training progress. The static budget uses a fixed $\rho_n = 0.4$, the midpoint of $[\rho_{\text{min}}, \rho_{\text{max}}]$.

Masking Budget	Training Progress				Avg. Pass@1
	0–25%	25–50%	50–75%	75–100%	
Static ($\rho_n = 0.4$)	20.41	13.56	9.46	8.44	45.91
Self-paced (ours)	**21.47**	**15.10**	**10.70**	**9.60**	**47.34**

Mask Overlap between Static and Self-Paced Budgets. To identify where this stronger signal comes from, we analyze the overlap of the masked prefixes between the two settings. The two settings mask a nearly identical total amount of prefixes, with a ratio of 0.998 between the total masking amounts of the self-paced and static budgets. However, the masked prefixes differ substantially, with a Jaccard similarity of 0.700 between the two masked token sets. This indicates that the benefit of self-paced scheduling comes from masking different prefixes, rather than masking more prefixes.

Budget Allocation across Decoding Positions. In Tab. 25, we report how the masking budget is allocated across decoding positions by measuring the average ρ_n across prefix-length bins. As the prefix grows, the full-context reverse KL divergence r_n decreases, indicating that later tokens become easier to imitate from the accumulated context. Accordingly, unlike the static budget, our self-paced scheduling gradually increases ρ_n along the trace (0.385 $\rightarrow$ 0.429), applying stronger masking to later positions where imitation becomes easier and the distillation signal weakens. As a result, the masked-context reverse KL divergence also increases along the trace (20.79 $\rightarrow$ 30.40), consistent with stronger perturbation at later decoding positions.

Table 25: **Budget allocation across decoding positions.** Each response token is assigned to a bin according to its prefix length (*i.e.*, the number of preceding response tokens), and statistics are averaged over 10K training traces. The full-context reverse KL divergence corresponds to r_n in Eq. (3), which is used for scheduling, and the masked-context reverse KL divergence corresponds to $\mathcal{L}_{\text{Distill}}$ in Eq. (1).

Prefix Length	0–1,024	1,024–2,048	2,048–3,072	3,072+
Full-context Reverse KL Div.	1.015	0.833	0.715	0.450
ρ_n (Static)	0.400	0.400	0.400	0.400
ρ_n (Self-paced, ours)	0.385	0.394	0.405	0.429
Masked-context Reverse KL Div.	20.79	23.65	25.11	30.40

Budget Allocation across Token Types. To verify whether the masking budget adapts to the availability of visual evidence, we classify each predicted token into function words, verbs, and content words using the spaCy part-of-speech tagger [12], and report per-type statistics in Tab. 26. Function words and verbs show lower visual attention, whereas content words rely more on the image. Accordingly, our self-paced scheduling applies weaker masking to function words and verbs ($\rho_n = 0.363$ and 0.357) and stronger masking to content words ($\rho_n = 0.410$). Moreover, the masked-context reverse KL divergence is also higher for content words, showing that masking concentrates its learning signal on tokens where visual evidence matters.

Table 26: **Budget allocation across predicted token types.** Each predicted token is classified by the spaCy part-of-speech tagger [12], and statistics are averaged over the 19K training traces. Visual attention denotes the average visual attention when predicting each token type.

Token Type	Function Words	Verbs	Content Words
Visual Attention	0.0144	0.0134	0.0152
ρ_n (Static)	0.400	0.400	0.400
ρ_n (Self-paced, ours)	0.363	0.357	0.410
Masked-context Reverse KL Div.	18.65	18.55	23.51

C.3 Statistics of Masked Prefix Positions

In this section, we analyze the relative position of masked prefixes with respect to the current token (*i.e.*, the distance from the current token to the masked prefix) over 19k teacher responses, as shown in Fig. 8. As described in the Implementation Details of Sec. 3.1 and Appendix B.4, we exclude the immediate previous prefix token from masking (*e.g.*, distance zero) to stabilize training. We observe that masking is more frequently applied to prefixes closer to the current token, resulting in a higher density of masked tokens in the recent context. As the distance from the current token increases, the masking frequency gradually decreases. This suggests that highly influential (*i.e.*, salient) reasoning prefixes are often located near the current token, and our masking strategy effectively targets these contexts.

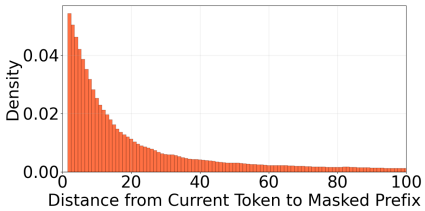


Figure 8: Masked prefix distance

C.4 Example of Inference

Fig. 9 illustrates the think-answer response of our Masking-KD compared with the undistilled student (*i.e.*, Qwen3-VL-2B-Thinking). Undistilled student produce perception errors, highlighted in red box, whereas ours show enhanced visual perception, highlighted in green box.

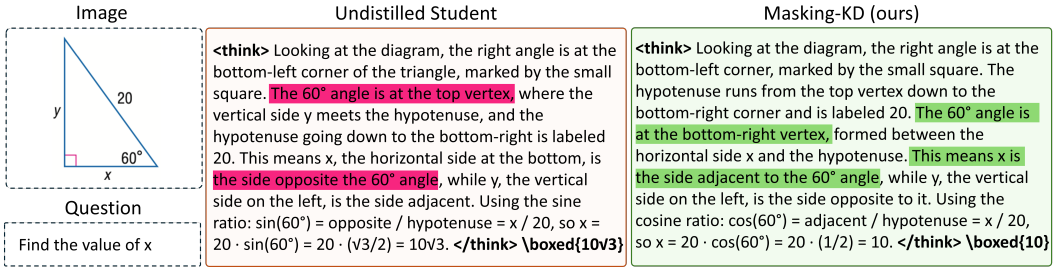


Figure 9: **Example of the Inference.**

C.5 More Comparison on Visual Attention Map

We provide additional comparisons on the visual attention map in Fig. 10. Compared with all compared methods, Masking-KD produces more focused attention on semantically relevant image regions, indicating that the student relies more on visual evidence throughout the reasoning process.

C.6 More Prediction Behavior of the Student during Distillation.

We illustrate the prediction behavior of the student during distillation without and with our salient reasoning-prefix mask in Fig. 6 of the main manuscript. In this section, we provide additional qualitative results across four types of reasoning problems: (a) math, (b) STEM, (c) table and (d) chart in Fig. 12. These examples further show that our masking strategy encourages the student to attend to relevant visual regions when predicting the current token, rather than relying solely on exposed textual reasoning prefixes.

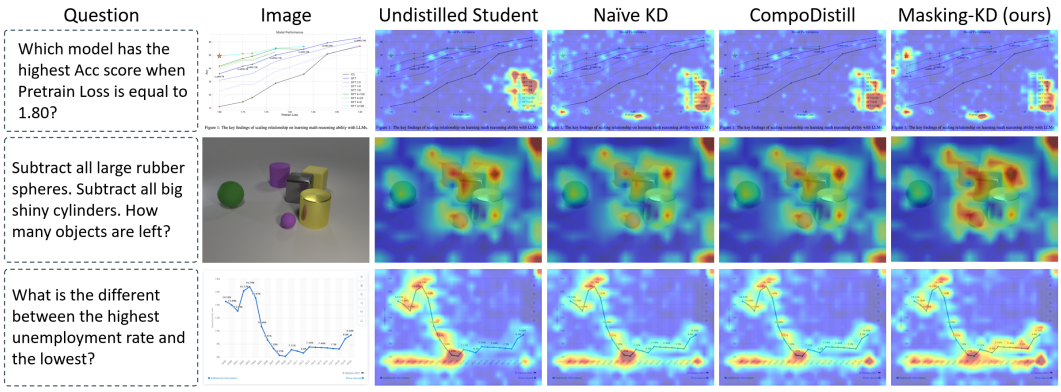


Figure 10: **More Comparison on Visual Attention Map.** We average the visual attention scores over the entire thinking trace.

D Additional Details

D.1 Details on Top-ρ Masking

To select salient prefixes to mask, we use a nucleus top-p style rule [11] (*i.e.*, top-ρ_n masking), as described in Eq. (4) of the main manuscript and rewritten below:

$$\sum_{j\in\mathcal{S}_n} \bar{\mathbf{A}}^{\text{resp}}_{n,j} \geq \rho_n, \quad \text{where } \bar{\mathbf{A}}^{\text{resp}}_{n,j} = \frac{\mathbf{A}^{\text{resp}}_{n,j}}{\sum_{k=1}^{n-1} \mathbf{A}^{\text{resp}}_{n,k}}. \tag{7}$$

In this section, we detail this top-ρ_n masking step-by-step.

For each n-th row in response-to-response attention map $\mathbf{A}^{\text{resp}}$, we first normalize its attention over preceding response tokens and sort the prefix tokens in descending order $\downarrow$ of attention, as follows:

$$\pi_n = \text{argsort}^{\downarrow}_{j<n}(\bar{\mathbf{A}}^{\text{resp}})_{n,j}, \quad \text{where } \bar{\mathbf{A}}^{\text{resp}}_{n,j} = \frac{\mathbf{A}^{\text{resp}}_{n,j}}{\sum_{k=1}^{n-1} \mathbf{A}^{\text{resp}}_{n,k}}. \tag{8}$$

We then collect the top-ranked prefix tokens until their cumulative attention mass reaches the self-paced cumulative ratio ρ_n:

$$\mathcal{S}_n = \{\pi_n(1), \ldots, \pi_n(k_n)\}, \quad \text{where } k_n = \min\left\{ K \,\middle|\, \sum_{i=1}^{K} \bar{\mathbf{A}}^{\text{resp}}_{n,\pi_n(i)} \geq \rho_n \right\}. \tag{9}$$

Here, $\bar{\mathbf{A}}^{\text{resp}}_{n,\pi_n(i)}$ denotes the attention assigned by response token y_n to the i-th highest-ranked prefix token under the ordering π_n. The resulting salient prefix set $\mathcal{S}_n$ for all $n = \{1, \ldots, N\}$ is then used to construct the salient reasoning-prefix mask $\tilde{\mathbf{M}}$ in Eq. (5).

D.2 Details on Self-Distillation

Our Masking-KD can operate under self-distillation settings, where the student serves as its own teacher. In self-distillation, the full-context student prediction is detached and used as the teacher target, while the masked-context student prediction is optimized to match it. This can be written by modifying the distillation loss in Eq. (1) as:

$$\mathcal{L}_{\text{Distill}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{y\in\mathcal{V}} p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \tilde{\mathbf{M}}) \log \frac{p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \tilde{\mathbf{M}})}{\text{StopGrad}\big(p_s(y \mid \mathbf{x}_v, \mathbf{x}_q, \mathbf{y}_{<n}, \mathbf{M})\big)}. \tag{10}$$

Here, StopGrad$(\cdot)$ denotes the stop-gradient operation, so the full-context branch serves as a fixed self-teacher target during optimization. All other hyperparameters and training recipes follow the knowledge distillation setting described in the Implementation Details of Sec. 3.1.

D.3 Instruction for Teacher-generated Response

To extract the teacher's think-answer trajectories from ViRL39k [29] dataset, we use the instruction illustrated in Fig. 11. The instruction is appended after the image and question to prompt the teacher model to generate think-answer trajectories.

Instruction
You first think through the reasoning process as an internal monologue, enclosed within <think> </think> tags. Then, provide your final answer enclosed within \boxed{}.

Figure 11: The instruction is used to prompt the teacher model to generate think-answer trajectories for distillation data.

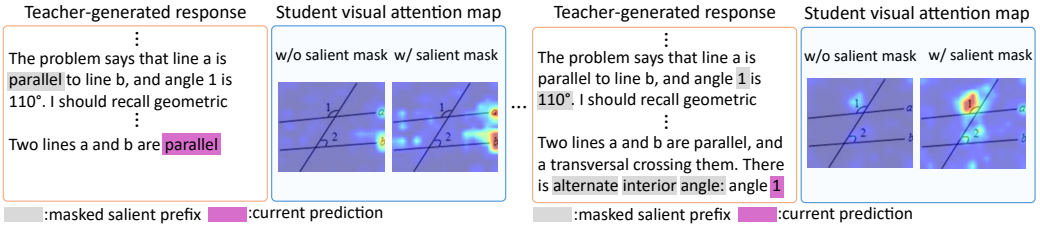
E Further Discussion

E.1 Limitation

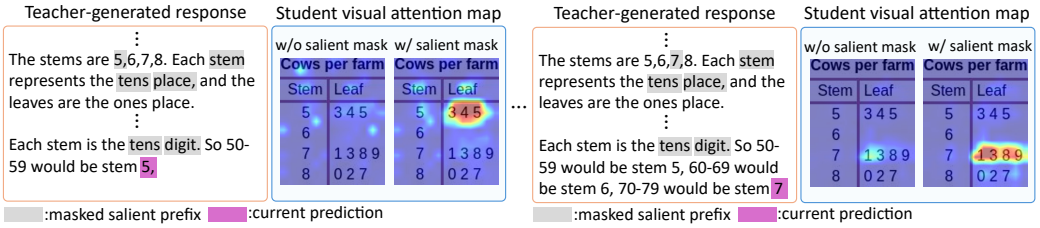
While Masking-KD introduces additional computational overhead and memory usage due to the auxiliary student forward pass and response-to-response attention map extraction, the overhead remains manageable in practice, as reported in Tab. 14 of Appendix A.4. In this work, we mainly focus on improving visually anchored reasoning during distillation rather than optimizing training efficiency. Further reducing the computational cost of salient prefix selection remains an important direction for future work.

E.2 Social Impact

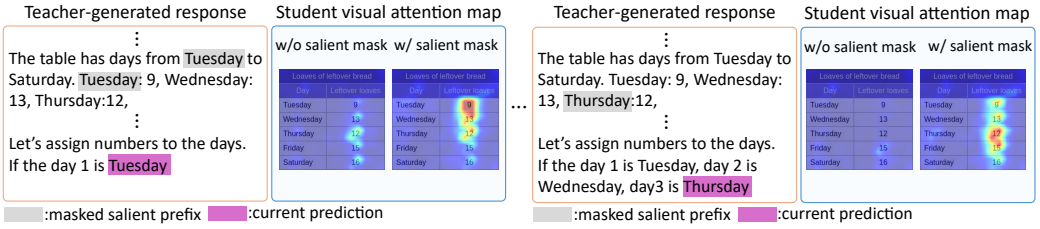
Our work improves the efficiency of think-answer VLMs by transferring reasoning capabilities to compact student models, which may help reduce deployment costs and broaden accessibility. However, like other VLMs, the distilled models may still inherit biases or generate incorrect outputs. Careful evaluation and responsible deployment are therefore important when applying these models in real-world settings.



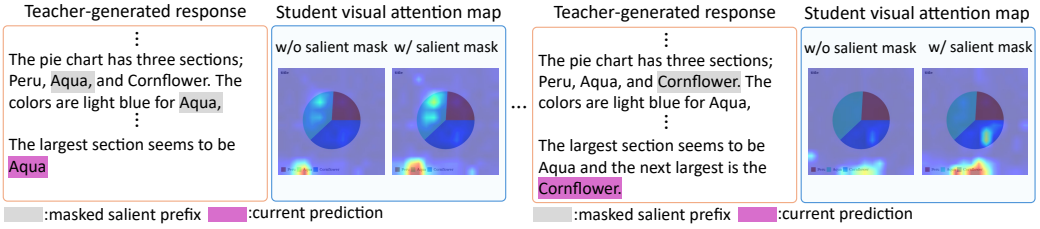
(a) math



(b) STEM



(c) table



(d) chart

Figure 12: **More Prediction Behavior of the Student during Distillation** without and with our salient reasoning-prefix mask across four types of reasoning problems: (a) math, (b) STEM, (c) table, and (d) chart.

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 contributions summarized in Sec. 1 are detailed.

   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: We discuss the limitations of this work in Appendix E.1.

   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: We do not present theoretical results or formal proofs in this paper.

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: We explain experimental setups in Sec. 3.1.

   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: Codes are included in the supplementary material Zip file. We use datasets that are publicly available. We will release our code on GitHub in the future.

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: We provide all necessary details to understand the results in Sec. 3.

   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: We report statistical significance of the experiments in Appendix A.5.

   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: We include the information on the computer resources in Sec. 3.1, Table 14.

   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: This work conducted with the NeurIPS Code of Ethics.

   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: We discuss social impacts in Sec. E.2.

    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: This paper does not pose such risks.

    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: We properly cite all assets and mention the license and terms of usage.

    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: We are releasing our code as a new asset, fully documented on GitHub, to complement the documentation in this paper.

    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 nor 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: This work does not involve human subjects, and therefore IRB approval is not required.

    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: [N/A].

Justification: LMs/VLMs are used as the subject of study rather than as a tool for developing the method.

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.