Model Guides You How to Draw: Adaptive Visual Gating for Unified Multimodal Reasoning

Wenxi Gao[1], **Guanxi Lu**[1], **Didi Zhu**[1], **Hao Mark Chen**[1]
Quan Deng[1,2], **Zhican Wang**[1], **Jiankang Deng**[1], **Hongxiang Fan**[1]
[1]Imperial College London [2]Tsinghua University
{w.gao25,guanxi.lu22,d.zhu,hao.chen20,q.deng25}@imperial.ac.uk
wangzhican1999@gmail.com, {j.deng16,hongxiang.fan}@imperial.ac.uk

Abstract

In interleaved reasoning, unified multimodal models (UMMs) generate both textual and visual steps as part of their intermediate reasoning traces. This capability has demonstrated great potential for multimodal mathematical reasoning. However, generating intermediate visual steps is not always helpful, as frequently triggering visual generation incurs substantial computational and memory overhead. Furthermore, our investigation shows that indiscriminate visual generation can even be harmful, as imperfect visual steps often introduce erroneous evidence that derails subsequent reasoning. In this work, we observe that the model's internal signals can indicate whether a visual step will benefit reasoning before visual generation completes. Specifically, we identify two key signals: *i) Generation Intent*, which reflects whether the model has a concrete textual plan for what to draw, and *ii) Visual Fidelity*, which measures whether the generation remains grounded in the original input image. Leveraging these signals, we propose ***AdaViG***, a training-free adaptive visual gating method for unified multimodal reasoning. ***AdaViG*** dynamically evaluates each triggered visual step at an early generation stage and aborts it when both signals are weak, thereby preventing misleading visual evidence from entering the reasoning trace while avoiding unnecessary computation. Comprehensive experiments show that ***AdaViG*** improves accuracy by up to 5.7% while reducing visual generation FLOPs by 25.0%–91.0% and wall-clock latency by 15.4%–45.6%. Our code is available at `https://github.com/wxgao8/AdaViG`.

1 Introduction

Recent advances in multimodal large language models (MLLMs) [1, 45] have shown promising performance in mathematical reasoning, particularly through multimodal chain-of-thought (MCoT) [39, 30, 40]. However, existing MCoT methods with text-only rationales [39, 30] are limited for vision math problems that require explicit visual construction as intermediate reasoning steps, such as drawing auxiliary lines or transforming diagrams. To overcome this limitation, MCoT has extended from text-only rationales to interleaved multimodal rationales [16, 36], where textual and visual steps are interleaved, enabling the model to reason with images.

Among various interleaved MCoT methods, tool-driven approaches [16, 36, 27, 32] operate over a fixed toolkit, such as zoom-in or cropping, limiting their flexibility. Code-based approaches [8, 13, 34, 41] can generate more diverse visual aids but depend on reliable code synthesis and rendering. Beyond these paradigms, intrinsic approaches [19, 12, 9, 18, 28] enable the model itself to generate visual steps, showing great potential for interleaved reasoning. This paradigm is primarily realized by unified multimodal models (UMMs) [7, 35, 4, 38], with fine-tuned variants such as Bagel-Zebra-CoT [18] and MathCanvas [28] natively interleaving visual generation into their reasoning traces.

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

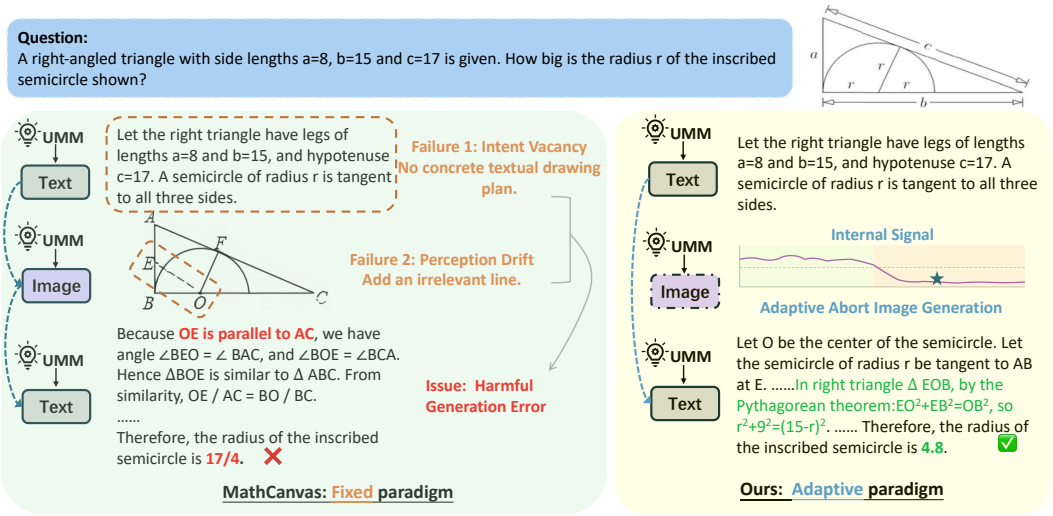


Figure 1: A representative harmful generation case from MathVision under the fixed MathCanvas paradigm exhibiting two failure modes (**left**), and how ***AdaViG*** adaptively aborts it (**right**).

However, whether a UMM emits a textual or visual step during reasoning is primarily dictated by the next action token. Such token-level triggering lacks an explicit mechanism to assess whether a visual step will actually benefit reasoning. We identify two systemic issues with this design: **(1) Harmful Generation Error.** Self-generated visual steps in UMMs can be redundant, incorrect, and misleading. Figure 1 illustrates that a harmful visual step misleads the subsequent text reasoning into a wrong answer, whereas the text-only counterfactual on the same sample reasons correctly. As shown in Figure 5(a), this reveals a significant *generation utility gap*: nearly 30% of self-generated visual steps actually degrade accuracy. **(2) Reasoning Inefficiency.** Frequent triggering of visual steps accounts for nearly 50% of overall inference latency and 80% of the extra KV-cache memory footprint, as shown in Figure 5(b).

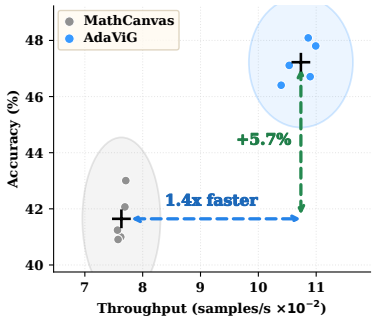


Figure 2: Accuracy and efficiency improvements over MathCanvas.

To address these limitations, we investigate how intrinsic visual generation hurts reasoning in UMMs. Our analysis reveals two recurring failure modes. **(1) Intent Vacancy**: the model draws without a clear textual plan. **(2) Perception Drift**: the generated image drifts from the original input diagram and misleads subsequent text reasoning steps. Motivated by these insights, we analyze early diffusion attention patterns during image generation and uncover two internal signals: ***Generation Intent***, which captures whether the model has a textual plan for what to draw, and ***Visual Fidelity***, which captures whether the generation remains grounded in the input diagram. Based on these signals, we propose ***AdaViG***, a training-free, plug-and-play adaptive visual gating method (Figure 1). ***AdaViG*** evaluates each visual step at an early generation stage and aborts potentially harmful ones before completion. This adaptive paradigm improves accuracy by up to 5.7% and achieves up to 1.4× higher throughput over MathCanvas, as shown in Figure 2.

Our contributions are as follows:

- We present the *generation utility gap* in unified multimodal reasoning, showing that certain intrinsic visual generations are not just redundant but can harm reasoning accuracy.
- We identify two internal signals during early diffusion that indicate whether an image generation is likely to help or hurt, and use these signals to build ***AdaViG***, a training-free adaptive visual gating method that aborts harmful generations early.
- Experiments across benchmarks show that ***AdaViG*** boosts accuracy by up to 5.7% while reducing generation FLOPs by 25.0%–91.0% and wall-clock latency by 15.4%–45.6%.

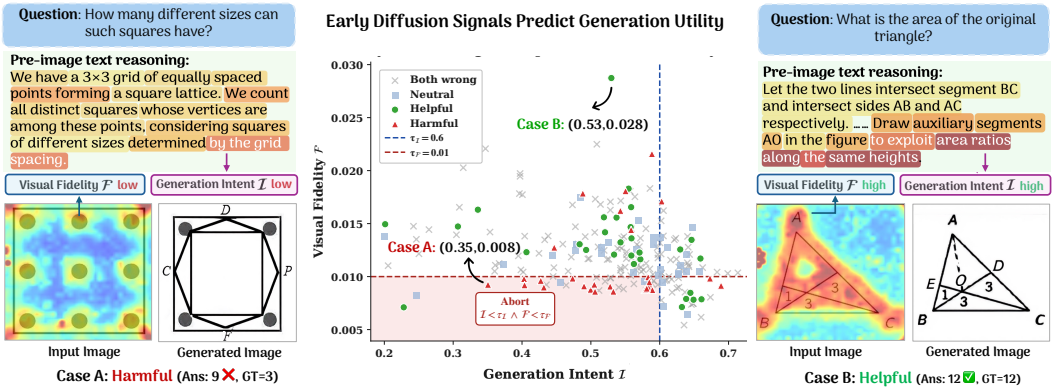


Figure 3: ***AdaViG* uses early diffusion signals to predict generation utility. Left**: A harmful trace has low $\mathcal{I}$ (e.g., no clear drawing plan in the text) and low $\mathcal{F}$ (e.g., the generated diagram drifts from the input image), placing the case in the abort region; the generated image distorts the input 3×3 lattice into a 2×2 pattern with extraneous polygons. **Middle**: The joint distribution of $(\mathcal{I}, \mathcal{F})$ at diffusion step $t^\star = 5$ of MathCanvas model on MathVision shows that *harmful* traces concentrate in the low-low region. **Right**: A helpful trace has high $\mathcal{I}$ (e.g., explicit plan to draw AO) and high $\mathcal{F}$ (e.g., heatmap traces the input triangle) and the gate retains the effective generation.

2 Related Work

Unified Multimodal Models. Unified multimodal models unify multimodal understanding and generation within a single architecture [7, 35, 4, 38, 44], often combining modality-specific components with a shared autoregressive, diffusion, or hybrid backbone. Prior work has improved UMM reasoning performance through post-training, including reinforcement learning [25, 31], self-generated supervision [29, 14], and supervised chain-of-thought (CoT)-style fine-tuning [28, 18]. In particular, UMM-based CoT methods exploit this unified text–image interface to produce interleaved multimodal reasoning traces. However, these methods do not assess whether a generated image will benefit subsequent reasoning, potentially incurring unnecessary generation costs and propagating misleading visual context. We address this limitation with a training-free gate that uses early diffusion attention during image generation to estimate image utility before full generation.

Interleaved Multimodal Mathematical Reasoning. Recent advances in multimodal large language models [1, 45] have shown strong performance on multimodal mathematical reasoning tasks [24, 33]. Textual rationales [39, 43, 30] enhance the reasoning ability of MLLMs by enabling multistage reasoning in a structured form, but they can be insufficient for problems whose solutions depend on spatial information. Interleaved multimodal rationales address this limitation by integrating visual evidence into the reasoning context [5, 11]. Existing interleaved reasoning methods can be grouped into three categories. Tool-driven methods [16, 36, 27, 32] invoke external visual operations, such as cropping, zooming, detection, or sketching, but are limited by a predefined operation space. Code-based methods [8, 13, 34] render auxiliary diagrams or plots from generated programs, but rely on accurate code synthesis and executable rendering. Intrinsic approaches [19, 18, 28] leverage the understanding and generation abilities of UMMs to generate visual thoughts directly.

Key Novelty of *AdaViG*. Complementary to these interleaved-reasoning paradigms, research has also explored efficient and adaptive interleaved multimodal reasoning. For example, [20, 22] use runtime signals to determine when and what visual context to insert, reducing redundant visual-token usage. Concurrently, [21, 2] inject visual and textual information into the latent reasoning process to improve inference efficiency. However, these approaches neither target intrinsic visual generation in UMMs with CoT nor assess whether generated visual context can harm subsequent reasoning.

3 Preliminaries

UMMs with CoT. Using MathCanvas [28] as an example, we formalize the reasoning process below. Let $\mathcal{M}$ denote a UMM with a Mixture-of-Transformers (MoT) design, in which separate understanding and generation experts, $\mathcal{E}_u$ and $\mathcal{E}_g$, share cross-modal self-attention layers. Given

a prompt (q, I^{in}) consisting of a question q and an optional input image I^{in}, $\mathcal{M}$ autoregressively produces an interleaved reasoning trace with N steps:

$$S = (s_1, s_2, s_3, \ldots, s_N), \tag{1}$$

where each step s_i is either a textual step E_i decoded through $\mathcal{E}_u$ or a visual step I_i^{gen} synthesized through $\mathcal{E}_g$ via a T-step diffusion process. The final element s_N is a text node containing the extracted answer $\hat{a}$. Let $\tau_i \in \{E, I^{\text{gen}}\}$ denote the type of s_i. The successor type is determined by

$$\tau_{i+1} = \begin{cases} E, & \tau_i = I^{\text{gen}} \quad \text{(text)} \\ I^{\text{gen}}, & \tau_i = E,\ a_i = \langle \texttt{img_start} \rangle \quad \text{(image)} \\ E, & \tau_i = E,\ a_i \neq \langle \texttt{img_start} \rangle \quad \text{(text)} \end{cases}, \tag{2}$$

where $a_i \sim P_{\mathcal{E}_u}(\cdot \mid s_{\leq i})$ denotes ... where $a_i \sim P_{\mathcal{E}_u}(\cdot \mid s_{<i})$ is the terminal action token of the current text node, which determines the type of the next node, and is sampled from the autoregressive decoding distribution of the $\mathcal{M}$.

Attention in UMMs. Image generation runs a T-step denoising diffusion inside the shared MoT, which applies causal attention on text tokens and bidirectional attention on vision tokens. Tokens fall into three pools: text ($\mathcal{S}_{\text{text}}$), ViT ($\mathcal{S}_{\text{vit}}$), and VAE ($\mathcal{S}_{\text{vae}}$). Every image ($I^{\text{in}}$ or I^{gen}) has both $\mathcal{S}_{\text{vit}}$ and $\mathcal{S}_{\text{vae}}$ tokens. At denoising step t, the noised VAE latent queries $\mathcal{Q}$ attend to all preceding keys $\mathcal{K}$. We define the segment attention mass at denoising step t as

$$m_t(\mathcal{Q}, \mathcal{K}) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \sum_{k \in \mathcal{K}} \alpha_t(q, k), \tag{3}$$

where $\alpha_t(q, k)$ denotes the head-averaged softmax attention weight from query q to key k.

4 Motivation

To understand the limitations of indiscriminate visual generation in UMMs, we first conduct an empirical analysis of the *generation utility gap*. Through this analysis, we characterize two systemic issues in accuracy and efficiency, which directly motivate our adaptive gating design.

Insight-1: Visual generations can sometimes harm reasoning accuracy. We denote the interleaved UMMs with CoT in Section 3 as $\mathcal{C}_{\text{gen}}$, where the action token a_i can trigger visual-step generation. To investigate the utility of visual steps, we construct a per-image counterfactual, denoted as $\mathcal{C}_{\text{abort}}$, by keeping the preceding reasoning context fixed and aborting the current visual step, while $\mathcal{C}_{\text{gen}}$ retains it. We then conduct contrastive experiments of $\mathcal{C}_{\text{gen}}$ and $\mathcal{C}_{\text{abort}}$ on MathVision using two different UMM-based CoT methods, Bagel-Zebra-CoT [18] and MathCanvas [28], and assign each visual step to one of three categories. (1) *Helpful*: $\mathcal{C}_{\text{gen}}$ produces the correct answer while $\mathcal{C}_{\text{abort}}$ is incorrect. *(2) Harmful*: $\mathcal{C}_{\text{gen}}$ generates the incorrect answer while $\mathcal{C}_{\text{abort}}$ is correct. *(3) Neutral*: both $\mathcal{C}_{\text{gen}}$ and $\mathcal{C}_{\text{abort}}$ generate the correct answer. As shown in Figure 4(a), on MathCanvas, 27.1% of samples are helpful, while a comparable 25.5% are harmful. Bagel-Zebra-CoT exhibits a similar pattern. We term this imbalance the *generation utility gap*: although intrinsic image generation can benefit some visual steps, its expected utility is close to zero because helpful and harmful generations largely offset each other.

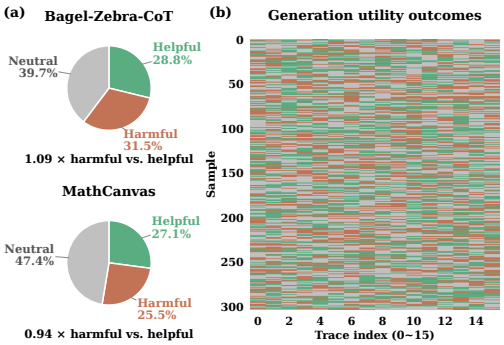


Figure 4: **Generation utility gap.** (a) Per-image counterfactual analysis reveals comparable rates of helpful and harmful visual generations. (b) Utility of visual generations varies across different reasoning traces for the same problem, indicating that harmfulness is not determined solely by the problem instance.

Insight-2: Harmfulness emerges from the evolving reasoning trace rather than from the problem alone. We further investigate what influences the utility gap by evaluating MathCanvas on MathVision with 16 independently sampled reasoning traces for each problem. Figure 4(b) visualizes the trace-level generation utility labels of all samples, where each cell denotes the utility label of one trace as helpful, harmful, or neutral. We find that 90.1% of samples exhibit at least two distinct utility labels across their 16 traces, and 40.5% of samples contain both a helpful trace and a harmful trace. Conversely, no sample is uniformly harmful, and only 9.9% of samples are uniformly neutral. These statistics imply that the harmfulness of image generation cannot be decided from the input query alone. Instead, it depends on the evolving partial reasoning trace and must be assessed adaptively at runtime when generation is about to occur. This motivates us to seek internal model signals that can support such adaptive decisions.

Performance and efficiency limitations. Based on the above observations, we identify two key limitations of existing methods. First, we observe the **Harmful Generation Errors**: as shown in Figure 5(a), approximately 14.6–22.6% of errors arise from generated visual content that either introduces incorrect information or adds redundant context. Such generations can mislead the reasoning process, particularly in long-context reasoning tasks. Second, we identify the **Reasoning Inefficiency**: as shown in Figure 5(b), image generation contributes more than 50% of the end-to-end latency and more than 80% of the KV-cache memory consumption across four benchmarks. These costs are incurred uniformly across all reasoning traces, including harmful and neutral traces where generation provides no accuracy gain. These findings motivate an adaptive mechanism that determines, for each reasoning trace, whether image generation is necessary, and makes this decision before most computation has been incurred.

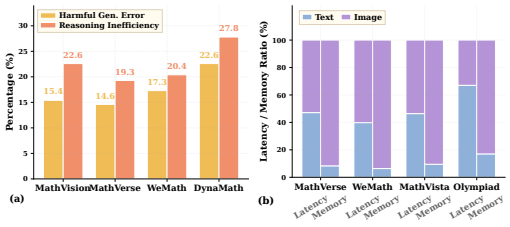


Figure 5: **Two systemic issues of UMMs with CoT.** (a) The proportion of harmful generation error and reasoning inefficiency, averaged over Bagel-Zebra-CoT and MathCanvas. (b) Image generation accounts for over 50% of end-to-end latency and 80% of KV-cache memory on average across four benchmarks.

5 Method

We propose ***AdaViG***, a training-free gating mechanism that decides whether a triggered image generation should proceed for each UMM reasoning trace. We first analyze how generated images mislead subsequent reasoning (Section 5.1), and then introduce two internal signals, *Generation Intent* (I) and *Visual Fidelity* (F). They capture the misleading potential at an early diffusion stage during image generation (Section 5.2). These two signals construct our adaptive gate (Section 5.3).

5.1 Generated Images Can Mislead Reasoning

Section 4 shows that self-generated images can reduce final accuracy. To understand this phenomenon, we analyze how this harm propagates through the reasoning trace. Figure 6 shows a representative harmful trace of MathCanvas [28] evaluated on WeMath [26]. The generated image distorts the input geometry due to *Intent Vacancy* and *Perception Drift* (Section 1) by producing an irregular non-closed polygon. Once appended to the reasoning context, this generated image receives higher attention ratio to the generated image than to the input image for **92%** of post-image text tokens, especially in the early reasoning span. The model then follows the spurious generated edge instead of the angle in the original question and produces an incorrect answer. This analysis reveals a critical failure mode: a harmful generated image can induce subsequent reasoning drift once it is consumed as key evidence. We provide a systematic statistical analysis in Appendix A.1.

5.2 Early Diffusion Attention Signals

As introduced in Section 3, image generation in UMMs operates as a diffusion process within a shared transformer. At diffusion step t, VAE queries attend to multiple token segments in the shared context, including the preceding text tokens $\mathcal{S}_{\text{text}}$, the input-image tokens $\mathcal{S}_{\text{vit}}^{\text{in}}$, $\mathcal{S}_{\text{vae}}^{\text{in}}$, and the current VAE latent tokens $\mathcal{S}_{\text{vae}}$. These segments have different semantic roles. The text tokens $\mathcal{S}_{\text{text}}$ provide

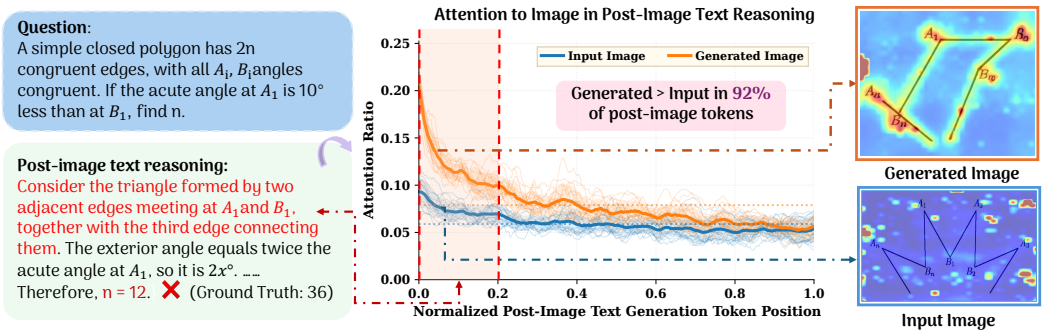


Figure 6: **Generated images can mislead later reasoning.** MathCanvas commits a *perception drift*, generating an irregular non-closed polygon that distorts the input. This induces an *attention drift*: post-image text attends more to the generated image (orange) than to the input (blue) in **92%** of tokens, with the largest gap in the early span. Heatmaps show attention concentrating on irrelevant edges, propagating erroneous evidence into subsequent reasoning and yielding a wrong answer.

the reasoning plan, specifying what the model intends to draw. The input image tokens $\mathcal{S}^{\text{in}}$ provide visual grounding, specifying what the generated image should remain faithful to. The VAE latent tokens $\mathcal{S}_{\text{vae}}$ capture the model's current self-denoising state.

Early diffusion attention weights can predict generation utility. Figure 3 visualizes this for the MathCanvas model on MathVision [33], contrasting helpful traces with harmful ones. We provide detailed attention visualizations in Appendix A.2. In helpful traces, VAE queries attend strongly to the preceding text $\mathcal{S}_{\text{text}}$ or the input image $\mathcal{S}^{\text{in}}$. In harmful traces, by contrast, they largely ignore both and focus disproportionately on the emerging image itself. This contrast motivates two quantitative statistics that respectively target *Intent Vacancy* and *Perception Drift*.

Let $m_t(\mathcal{Q}_{\text{vae}}, \mathcal{S})$ denote the attention mass from VAE queries $\mathcal{Q}_{\text{vae}}$ to a token segment $\mathcal{S}$ at diffusion step t. We define two early diffusion statistics at an early decision step $t^\star$.

Generation Intent. The first metric measures the relative balance between preceding text reasoning $\mathcal{S}_{\text{text}}$ and input image $\mathcal{S}^{\text{in}} = \mathcal{S}^{\text{in}}_{\text{vit}} \cup \mathcal{S}^{\text{in}}_{\text{vae}}$. We define *Generation Intent*:

$$\mathcal{I} = \frac{m_{t^\star}(\mathcal{Q}_{\text{vae}}, \mathcal{S}_{\text{text}})}{m_{t^\star}(\mathcal{Q}_{\text{vae}}, \mathcal{S}_{\text{text}}) + m_{t^\star}(\mathcal{Q}_{\text{vae}}, \mathcal{S}^{\text{in}})}. \tag{4}$$

A high $\mathcal{I}$ indicates that the emerging image is conditioned predominantly on the text reasoning plan among the two problem-specific sources, corresponding to text plan-driven generation and reducing the risk of *Intent Vacancy*. In contrast, a low $\mathcal{I}$ indicates dominance by the input image evidence. The absence of explicit textual guidance can lead to unconstrained elaboration of the input diagram and increase the risk of *Perception Drift*.

Visual Fidelity. The second metric captures whether the generation maintains sufficient attention to the input image. We define *Visual Fidelity*:

$$\mathcal{F} = m_{t^\star}\left(\mathcal{Q}_{\text{vae}}, \mathcal{S}^{\text{in}}_{\text{vit}}\right). \tag{5}$$

A low $\mathcal{F}$ suggests that the generation is weakly related to the original input diagram and may drift toward irrelevant or misleading visual content, corresponding to *Perception Drift*.

5.3 Adaptive Visual Gating (*AdaViG*)

Since $\mathcal{I}$ and $\mathcal{F}$ capture complementary grounding sources, neither signal alone suffices for gating. A visual step with low $\mathcal{I}$ can still be helpful if it strongly focuses on the input image, while one with low $\mathcal{F}$ can still be effective when guided by an explicit textual plan. Thus, we adopt a conservative rule that aborts generation only when both signals are weak:

$$g(\mathcal{I}, \mathcal{F}) = \begin{cases} \text{ABORT}, & \mathcal{I} < \tau_{\mathcal{I}} \ \wedge \ \mathcal{F} < \tau_{\mathcal{F}}, \\ \text{CONTINUE}, & \text{otherwise.} \end{cases} \tag{6}$$

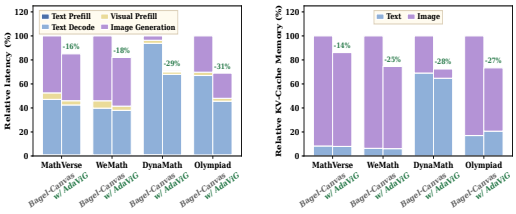


Figure 7: Relative latency and KV-cache memory breakdown across benchmarks.

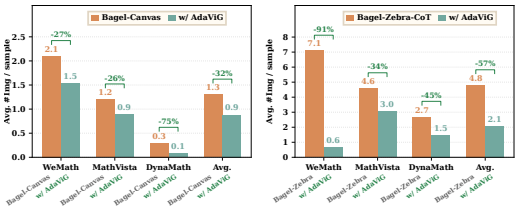


Figure 8: Reduction in the average number of generated images per sample.

If the gate returns CONTINUE, diffusion proceeds normally. If it returns ABORT, ***AdaViG*** stops denoising at step $t^\star$, discards the partial VAE latents, and resets to a text node. We provide the detailed algorithm in Appendix A.3.

Computational overhead. ***AdaViG*** intervenes only when image generation is triggered. The gate itself adds negligible overhead: $\mathcal{I}$ and $\mathcal{F}$ are computed from attention values that are already produced during the first $t^\star$ diffusion steps, requiring only a sum reduction over selected key segments. When the gate aborts, nearly $(1 - t^\star/T)$ of the diffusion cost is saved; when it continues, the only overhead is the attention reduction at step $t^\star$.

6 Experiments

6.1 Experimental Setup

Implementations. We evaluate ***AdaViG*** on two UMM-based CoT models with intrinsic image generation: Bagel-Zebra-CoT [18] and Bagel-Canvas from the MathCanvas framework [28], both based on the Bagel-7B [7] architecture. We set the early diffusion step to $t^\star = 5$, and the thresholds $(\tau_{\mathcal{I}}, \tau_{\mathcal{F}})$ to $(0.5, 0.04)$ for Bagel-Zebra-CoT and $(0.6, 0.01)$ for Bagel-Canvas. These are tuned on MathVision `testmini` (304 samples). The effect of the thresholds is detailed in Appendix A.9. All experiments run on NVIDIA H200 GPUs with 141 GB of memory.

Baselines. We compare ***AdaViG*** against three types of baselines: (1) General MLLMs, including GPT-4o [17], Gemini-2.5-Pro [6], Qwen2.5-VL-7B [1], and InternVL3-8B [45]; (2) UMMs, including Emu3-Chat-8B [35], Janus-Pro-7B [4], MUSE-VL-7B [38], and Bagel-7B [7]; and (3) **CoT-Prompted UMMs**, applying either text-only CoT [37] or interleaved multimodal CoT (ICoT) [11].

Benchmarks. We evaluate ***AdaViG*** on five math reasoning benchmarks: MathVista `testmini` [24], MathVerse `testmini` [42], OlympiadBench [15], WeMath [26], and DynaMath [46]. We use the GPS category of MathVista `testmini` and the Math subset of OlympiadBench.

6.2 Main Results

Table 1 presents the accuracy and efficiency gains, where efficiency is measured by the reduction in FLOPs for image generation, as detailed in Appendix A.4. Overall, ***AdaViG*** consistently improves accuracy and image-generation efficiency across models and datasets. ***AdaViG*** reaches an average performance of 50.5% under the same BAGEL-7B backbone, significantly outperforming existing UMM reasoning methods such as CoT [37] (42.1%) and ICoT [11] (43.0%). Notably, on DynaMath with Bagel-Canvas, ***AdaViG*** improves accuracy by 5.7% while reducing image-generation FLOPs by 75%. These results show that ***AdaViG*** improves intrinsic image generation in UMMs with CoT, outperforming methods whose reasoning chains rely only on text (*CoT*) or on visual evidence extracted from the original input image (*ICoT*). We further provide case studies in Appendix A.5.

Table 2 shows that ***AdaViG*** also brings consistent end-to-end latency reductions, positively correlated with the reduction in image generations. Bagel-Zebra-CoT achieves larger latency reductions, up to 45.6%, while Bagel-Canvas achieves 15.4%–30.8%. Since latency also depends on the length of the remaining and additional text reasoning process, its improvement can be lower than the reduction in image generations. We further decompose latency and KV-cache memory into text and image

Table 1: **Main results on math reasoning benchmarks.** The best accuracy (%) results among UMMs *w/ CoT* are in **bold**, and the second best are underlined. Orange indicates accuracy gains; blue indicates efficiency gains, measured by the reduction (%) in FLOPs for image generation.

Method		DynaMath	WeMath	MathVerse	MathVista	Olympiad	Avg.
General MLLMs							
GPT-4o		56.3	45.8	50.8	60.0	33.0	49.2
Gemini-2.5-Pro		66.8	78.0	76.9	80.9	42.1	68.9
Qwen2.5-VL-7B		49.8	35.3	45.7	72.6	22.0	45.1
InternVL3-8B		37.1	34.8	39.8	70.8	21.5	40.8
UMMs							
Emu3-Chat-8B		24.6	20.5	15.5	47.6	10.2	23.7
Janus-Pro-7B		20.3	16.3	15.9	42.8	9.7	21.0
MUSE-VL-7B		30.8	30.2	39.1	51.3	15.0	33.3
Bagel-7B		38.2	36.2	42.6	68.8	22.7	41.7
UMMs w/ CoT							
Bagel-7B *w/ CoT*		38.6	37.1	43.0	68.8	23.0	42.1
Bagel-7B *w/ ICoT*		39.5	37.0	45.8	70.0	22.7	43.0
Bagel-Zebra-CoT		39.0	34.5	48.2	81.0	16.7	43.9
*w/ **AdaViG (Ours)***	**Acc.** ↑	43.5 (+4.5)	38.1 (+3.6)	51.3 (+3.1)	**83.4** (+2.4)	18.9 (+2.2)	47.0 (+3.1)
	FLOPs ↓	-45%	-91%	-75%	-34%	-40%	-57%
Bagel-Canvas		42.3	35.3	55.8	77.9	24.0	47.1
*w/ **AdaViG (Ours)***	**Acc.** ↑	**48.0** (+5.7)	**38.8** (+3.5)	**58.4** (+2.6)	81.3 (+3.4)	**26.2** (+2.2)	**50.5** (+3.4)
	FLOPs ↓	-75%	-27%	-25%	-26%	-37%	-32%

Table 2: **End-to-end inference latency (seconds per sample).** The results are measured on a single H200 GPU and averaged across each benchmark. Blue indicates the reduction in latency.

Method	DynaMath	WeMath	MathVerse	MathVista	Olympiad
Bagel-Zebra-CoT	37.4	41.6	53.2	52.7	24.9
*w/ **AdaViG (Ours)***	25.7 (-31.3%)	22.6 (-45.6%)	31.7 (-40.0%)	39.8 (-24.5%)	17.6 (-29.3%)
Bagel-Canvas	13.2	13.1	11.5	10.4	5.72
*w/ **AdaViG (Ours)***	9.4 (-28.9%)	10.8 (-17.6%)	9.7 (-15.7%)	8.8 (-15.4%)	3.96 (-30.8%)

components (Figure 7). Both components decrease, suggesting that suppressing harmful image generations also reduces the cost of the text reasoning that would otherwise be conditioned on them.

Discussion on different models. We observe that ***AdaViG*** achieves larger accuracy gains (3.4%) on Bagel-Canvas, while bringing larger efficiency gains (57%) on Bagel-Zebra-CoT. As shown in Figure 8, Bagel-Canvas generates an average of 1.3 images per sample reasoning trace, whereas Bagel-Zebra-CoT generates 4.8 images. This indicates that Bagel-Canvas produces more targeted image generations, as its supervised fine-tuning (SFT) on math tasks equips it with stronger capabilities. As a result, ***AdaViG*** tends to improve accuracy more on stronger models by aborting harmful image generations, while improving efficiency more on weaker models by suppressing redundant ones.

Robustness across generation frequencies. As shown in Figure 8, Bagel-Canvas exhibits substantial variation in image-generation frequency across benchmarks: WeMath triggers frequent image generation (2.1 images per sample), MathVista a moderate amount (1.2), and DynaMath sparse image generation (0.3). Across all the settings, ***AdaViG*** consistently improves performance by +3.5%, +3.4%, and +5.7%, respectively, demonstrating robustness to varying image-generation frequencies.

Generalization across UMM architectures. We further evaluate ***AdaViG*** on two non-BAGEL UMMs, Anole-Zebra-CoT [18] and UniRL [25], representing autoregressive and Show-o-based architectures. For Anole-Zebra-CoT, the gating decision is made after an early prefix of visual

Table 3: **Generalization across UMM architectures.** Accuracy (%) and FLOPs reduction.

Method	DynaMath	MathVerse	WeMath	FLOPs
Anole-Zebra-CoT	31.3	23.8	26.4	–
+ ***AdaViG***	**33.4**	**25.6**	**28.8**	45%
UniRL	34.2	26.5	28.7	–
+ ***AdaViG***	**36.0**	**28.5**	**30.6**	34%

Table 4: **Generalization beyond math reasoning.** Accuracy (%) and average FLOPs reduction.

Benchmark	Baseline	AdaViG	FLOPs
MMBench [23]	65.7	**68.0**	
MMStar [3]	53.5	**56.5**	53%
BLINK [10]	45.6	**48.2**	

Table 5: Performance on Bagel-Canvas under gating methods with different ($\mathcal{I}$,$\mathcal{F}$) combinations.

Method	Avg. Drop	Performance		
		MathVista	WeMath	DynaMath
Bagel-Canvas	–	77.9	35.3	42.3
w/ $\mathcal{I}$ only	55%	78.8	36.2	46.1
w/ $\mathcal{F}$ only	31%	79.4	36.5	45.4
w/ $\mathcal{I} \vee \mathcal{F}$	61%	75.1	32.7	41.2
w/ ***AdaViG***	43%	**81.3**	**38.8**	**48.0**

Table 6: Comparison with alternative image generation aborting strategies on Bagel-Canvas.

Method	Drop Rate	Performance		
		MathVista	WeMath	DynaMath
AdaViG	43%	**81.3**	**38.8**	**48.0**
Random	25%	78.2	34.6	44.1
	50%	77.4	35.6	42.9
Text-only	100%	76.9	34.1	42.7
DAP-ICoT [22]	12%	78.4	35.4	44.3

tokens. As shown in Table 3, ***AdaViG*** improves reasoning accuracy by 1.8%–2.4% while reducing image-generation FLOPs by 34–45%, showing its generalizability across architectures.

Generalization beyond mathematical reasoning. We further evaluate ***AdaViG*** on three general multimodal understanding benchmarks. As shown in Table 4, ***AdaViG*** improves accuracy by 2.3%, 3.0%, and 2.6%, respectively, while reducing average image-generation FLOPs by 53%. These results demonstrate that ***AdaViG*** generalizes beyond mathematical reasoning tasks.

Implications for improving UMM reasoning ability. Applying ***AdaViG*** to Bagel-Zebra-CoT, without math-specific tuning, improves its accuracy from 43.9 to 47.0, nearly matching the 47.1 accuracy of the math-tuned Bagel-Canvas and recovering about 95% of the gain from specialized math training. This result suggests two potential directions for improving UMM reasoning: (1) using our gating method to filter interleaved training data and retain effective reasoning traces, and (2) using our metric as a reward signal during reinforcement learning.

6.3 Ablation Studies

Performance of different gating methods. Table 5 evaluates Bagel-Canvas with different image generation abort criteria based on $\mathcal{I}$ and $\mathcal{F}$. Results on Bagel-Zebra-CoT are reported in Appendix A.8. The variants "w/ $\mathcal{I}$" and "w/ $\mathcal{F}$" use a single weak signal for abortion, while "w/ $\mathcal{I} \vee \mathcal{F}$" aborts when either signal is weak. Single signal gates improve over Bagel-Canvas but remain suboptimal, as the two signals capture different failure modes. The $\mathcal{I} \vee \mathcal{F}$ variant removes the most generations (61%) but reduces accuracy by 2.2% on average. In contrast, ***AdaViG*** aborts only when both signals are weak and achieves the best performance with an average accuracy gain of 4.2% and a moderate drop rate of 43%. This indicates that gating with both signals better balances accuracy and efficiency by preserving helpful image generations while aborting harmful ones.

Comparison with alternative aborting strategies. Table 6 compares ***AdaViG*** with random image generation aborting, text-only reasoning, and DAP-ICoT [22] on Bagel-Canvas. Both random aborting and the text-only methods underperform ***AdaViG*** by 3.9% and 4.8% on average, indicating that naive approaches of removing image generations are less effective. Since prior efficiency methods do not target UMMs with CoT, we adapt DAP-ICoT [22] in our settings. It yields a modest average gain of 0.9% over Bagel-Canvas, likely because it relies on textual confidence to determine whether to insert visual steps, which is suboptimal for UMM-based reasoning. In contrast, ***AdaViG*** achieves the best performance, suggesting that its improvement does not come from merely reducing image generations, but from selectively suppressing harmful ones. Additional random-drop experiments

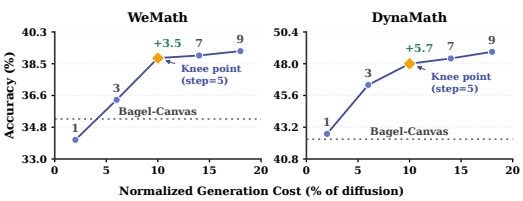


Figure 9: Early aborting step trade-off.

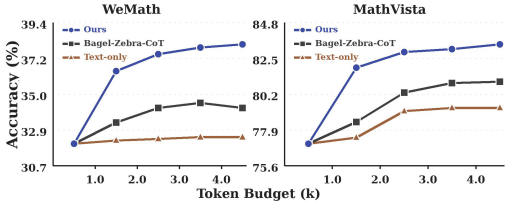


Figure 10: Accuracy under varying token budget.

and stronger text-only comparisons are provided in Appendices A.6 and A.7, respectively. Results on Bagel-Zebra-CoT are provided in Appendix A.8.

Selection of early aborting step $t^\star$. We vary the early diffusion step for image generation abortion, as shown in Figure 9. The normalized generation cost is t/T, where t is the decision step and $T = 50$ is the total number of diffusion steps. At $t^\star = 5$, ***AdaViG*** improves accuracy by 4.6% on average, while consuming only 10% of the diffusion steps before the gating decision. The results reveal a trade-off between accuracy and efficiency: decisions at later steps provide stronger gating capability but incur higher generation cost, highlighting the importance of choosing an appropriate early aborting step.

Performance across different token budgets. We compare ***AdaViG*** with Bagel-Zebra-CoT and its text-only variant under different token budgets, as shown in Figure 10. With a tight budget, the methods perform similarly because few image generations can be triggered. As the budget increases, Bagel-Zebra-CoT first benefits from more interleaved visual steps, but later degrades as harmful image generations become more likely. In contrast, ***AdaViG*** keeps improving by aborting low-utility generations, outperforming Bagel-Zebra-CoT by 3.2% on average at the largest token budget. These results suggest that visual gating is important when scaling intrinsic interleaved reasoning with larger token budgets.

Limitations. ***AdaViG*** improves unified multimodal reasoning by reducing harmful intermediate image generations using internal attention signals. However, its benefits may be limited on reasoning chains where intermediate visual generation is rarely triggered. Beyond deciding whether to generate a visual aid, the gate does not directly improve the quality of the generated visual tokens. Addressing this limitation therefore requires complementary techniques that enhance visual generation itself, particularly when the underlying image generation capability is weak and high-fidelity visual support is necessary. More broadly, when errors arise from limitations in the underlying model's reasoning capability, they cannot be addressed through the gating alone. Looking forward, we plan to incorporate Generation Intent and Visual Fidelity into CoT supervised fine-tuning as training-time supervision signals, rather than using them only as inference-time gating criteria.

7 Conclusion

This paper investigates when intrinsic visual generation helps or hurts unified multimodal math reasoning. We show that the fixed generation paradigm in current UMMs with CoT can introduce harmful visual evidence and unnecessary costs. Motivated by this, we propose ***AdaViG***, a training-free adaptive visual gating method that uses early diffusion attention to measure *Generation Intent* and *Visual Fidelity*, and aborts visual generation at an early stage when both signals are weak. Experiments show that ***AdaViG*** improves accuracy while reducing latency. These results highlight the importance of utility-aware visual generation for future UMMs with CoT training.

Acknowledgments and Disclosure of Funding

The support of the UKRI (Grant 256) and Microsoft is gratefully acknowledged. The authors declare that there are no competing interests in this work.

References

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

[2] Chao Chen, Zhixin Ma, Yongqi Li, Yupeng Hu, Yinwei Wei, Wenjie Li, and Liqiang Nie. Reasoning in the dark: Interleaved vision-text reasoning in latent space. *arXiv preprint arXiv:2510.12603*, 2025.

[3] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

[4] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

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

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

[7] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[8] Chengqi Duan, Kaiyue Sun, Rongyao Fang, Manyuan Zhang, Yan Feng, Ying Luo, Yufang Liu, Ke Wang, Peng Pei, Xunliang Cai, et al. Codeplot-cot: Mathematical visual reasoning by thinking with code-driven images. *arXiv preprint arXiv:2510.11718*, 2025.

[9] Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*, 2025.

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

[11] Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. Interleaved-modal chain-of-thought. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19520–19529, 2025.

[12] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, et al. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025.

[13] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14953–14962, 2023.

[14] Ruiyan Han, Zhen Fang, XinYu Sun, Yuchen Ma, Ziheng Wang, Yu Zeng, Zehui Chen, Lin Chen, Wenxuan Huang, Wei-Jie Xu, et al. Unicorn: Towards self-improving unified multimodal models through self-generated supervision. *arXiv preprint arXiv:2601.03193*, 2026.

[15] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

[16] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

[17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[18] Ang Li, Charles Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, et al. Zebra-cot: A dataset for interleaved vision language reasoning. *arXiv preprint arXiv:2507.16746*, 2025.

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

[20] Xiping Li and Jianghong Ma. Aimcot: Active information-driven multimodal chain-of-thought for vision-language reasoning. *arXiv preprint arXiv:2509.25699*, 2025.

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

[22] Xu Liu, Yongheng Zhang, Qiguang Chen, Yao Li, Sheng Wang, and Libo Qin. Let's think with images efficiently! an interleaved-modal chain-of-thought reasoning framework with dynamic and precise visual thoughts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32213–32221, 2026.

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

[24] 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. *arXiv preprint arXiv:2310.02255*, 2023.

[25] Weijia Mao, Zhenheng Yang, and Mike Zheng Shou. Unirl: Self-improving unified multimodal models via supervised and reinforcement learning. *arXiv preprint arXiv:2505.23380*, 2025.

[26] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Yifan Zhang, Xiao Zong, Yida Xu, Muxi Diao, Zhimin Bao, Chen Li, and Honggang Zhang. We-math: Does your large multimodal model achieve human-like mathematical reasoning?, 2024.

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

[28] Weikang Shi, Aldrich Yu, Rongyao Fang, Houxing Ren, Ke Wang, Aojun Zhou, Changyao Tian, Xinyu Fu, Yuxuan Hu, Zimu Lu, et al. Mathcanvas: Intrinsic visual chain-of-thought for multimodal mathematical reasoning. *arXiv preprint arXiv:2510.14958*, 2025.

[29] Zihan Su, Hongyang Wei, Kangrui Cen, Yong Wang, Guanhua Chen, Chun Yuan, and Xiangxiang Chu. Generation enhances understanding in unified multimodal models via multi-representation generation. *arXiv preprint arXiv:2601.21406*, 2026.

[30] Omkar Thawakar, Dinura Dissanayake, Ketan Pravin More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Ilmuz Zaman Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. Llamav-o1: Rethinking step-by-step visual reasoning in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24290–24315, 2025.

[31] Fu-Yun Wang, Han Zhang, Michael Gharbi, Hongsheng Li, and Taesung Park. Unirl-zero: Reinforcement learning on unified models with joint language model and diffusion model experts. *arXiv preprint arXiv:2510.17937*, 2025.

[32] Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025.

[33] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

[34] Ke Wang, Junting Pan, Linda Wei, Aojun Zhou, Weikang Shi, Zimu Lu, Han Xiao, Yunqiao Yang, Houxing Ren, Mingjie Zhan, et al. Mathcoder-vl: Bridging vision and code for enhanced multimodal mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2505–2534, 2025.

[35] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[36] Yikun Wang, Siyin Wang, Qinyuan Cheng, Zhaoye Fei, Liang Ding, Qipeng Guo, Dacheng Tao, and Xipeng Qiu. Visuothink: Empowering lvlm reasoning with multimodal tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21707–21719, 2025.

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

[38] Rongchang Xie, Chen Du, Ping Song, and Chang Liu. Muse-vl: Modeling unified vlm through semantic discrete encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24135–24146, 2025.

[39] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2087–2098, 2025.

[40] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*, 2024.

[41] Jieyu Zhang, Le Xue, Linxin Song, Jun Wang, Weikai Huang, Manli Shu, An Yan, Zixian Ma, Juan Carlos Niebles, Silvio Savarese, et al. Provision: Programmatically scaling vision-centric instruction data for multimodal language models. *arXiv preprint arXiv:2412.07012*, 2024.

[42] 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 *European Conference on Computer Vision*, pages 169–186. Springer, 2024.

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

[44] Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

[45] 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.

[46] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*, 2024.

A Appendix

A.1 Detailed Analysis on How Generated Images Mislead Reasoning

Section 5.1 shows a single case where a generated image attracts more attention than the input image during later text reasoning. As shown in Figure 11, we verify the generalization of this phenomenon on WeMath, MathVista, and MathVerse by averaging the attention ratio across all samples.

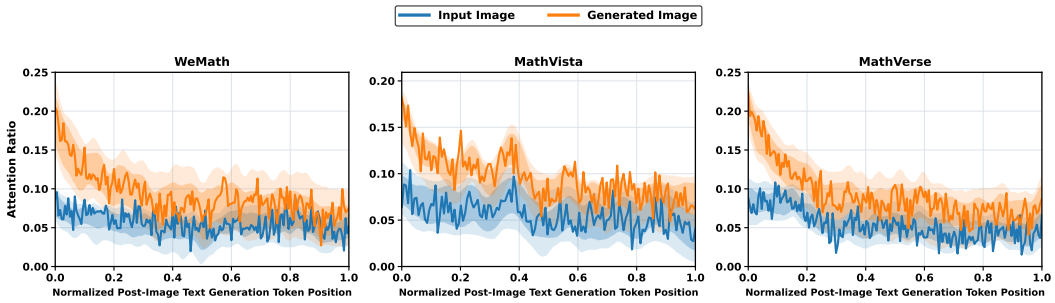


Figure 11: Generated images attract more attention than input images during later text reasoning across benchmarks.

A.2 Attention Visualization of Helpful and Harmful Generation Cases

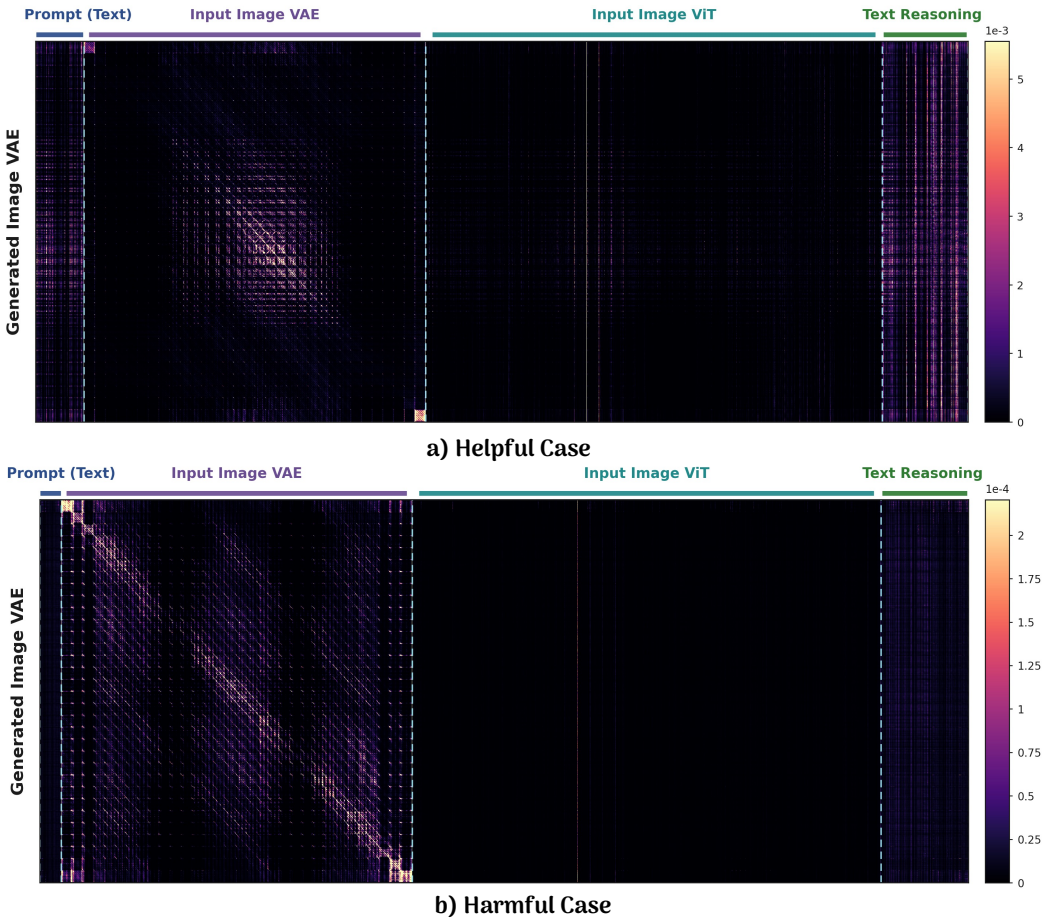


Figure 12: Helpful case attends more to text reasoning and input image ViT than harmful case.

A.3 *AdaViG* Algorithm

Algorithm 1 ***AdaViG***: Adaptive Visual Gating

Require: Context c, decision step $t^\star$, thresholds $(\tau_{\mathcal{I}}, \tau_{\mathcal{F}})$
Predict next action token a
if $a \neq \langle \texttt{img_start} \rangle$ **then**
    $\tau_i = E$; Continue text decoding
else
    $\tau_i = I^{\text{gen}}$; Run diffusion for $t^\star$ steps
    Compute $\mathcal{I}$ and $\mathcal{F}$ from early VAE attention
    if $\mathcal{I} < \tau_{\mathcal{I}}$ **and** $\mathcal{F} < \tau_{\mathcal{F}}$ **then**
        Abort diffusion; discard partial VAE latents
        $\tau_i = E$; Reset to text node
    else
        Continue diffusion; append generated image
    end if
end if

A.4 Calculation of FLOPs

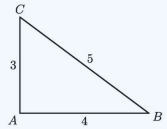
We estimate image generation FLOPs from the number of executed diffusion steps. A full image generation requires T denoising steps, while an aborted generation under ***AdaViG*** only executes the first $t^\star$ steps. Thus, each aborted generation saves approximately $(T - t^\star)/T$ of its image-generation FLOPs. Given an abort rate r, the overall image generation FLOPs reduction is estimated as

$$\Delta_{\text{FLOPs}} = r\left(1 - \frac{t^\star}{T}\right). \tag{7}$$

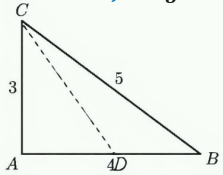
We use $t^\star = 5$ and $T = 50$ in all experiments, so an aborted generation saves about 90% of its image generation FLOPs.

A.5 Case Studies

Question:
In the figure below, choose point D on BC so that △ACD and △ABD have equal perimeters. What is the area of △ ABD?
Options: A. $\frac{3}{4}$, B. $\frac{3}{2}$, C. 2, D. $\frac{12}{5}$, E. $\frac{5}{2}$

Input Image:


Solution:
Text: Let BD = x, so CD = BC - BD = 5 - x. The perimeters are: P_{ABD} = AB + AD + BD = 4 + AD + x. P_{ACD} = AC + AD + CD = 3 + AD + (5 - x) = 8 + AD - x. Equal perimeters give x = 2.
Visual Gating: $\mathcal{I}$ = 0.42 $\mathcal{F}$ = 0.007 ⟶ **Abort Generation**
(Ours) Text: The area of △ABC is $\frac{1}{2}AB \cdot AC = 6$. Also, the area of △ABD equals $\frac{BD}{BC}$ times △ABC. So the area of △ABD is $\frac{2}{5} \cdot 6 = \frac{12}{5}$. Therefore, **the correct option is D.** ✅

(MathCanvas) Image


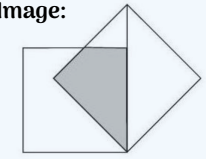
Text: Since △ ACD and △ ABD have equal perimeters and share side AD, the remaining sides satisfy $AC = AB$. So △ ABC is isosceles with base AB. Drop $AD \perp BC$ at D. Then $BD = DC = \frac{BC}{2}$.
By the Pythagorean theorem: $AB^2 = AD^2 + BD^2 \implies AD = 2\sqrt{3}$. Therefore: $S_{\triangle ABD} = \frac{1}{2} \cdot AB \cdot AD = \frac{1}{2} \cdot 4 \cdot 2\sqrt{3} = 4\sqrt{3}$. Since the options are unitless, $4\sqrt{3}$ corresponds to **option A.** ❌

Figure 13: Case study between our method and MathCanvas.

Question:
Two squares with side 1 have a common vertex, and the edge of one of them lies along the diagonal of the other. What is the area of the overlap between the squares?

Option: A. $\sqrt{2}-1$, B. $\frac{\sqrt{2}}{2}$, C. $\frac{\sqrt{2}+1}{2}$, D. $\sqrt{2}+1$, E. $\sqrt{3}-\sqrt{2}$

Input Image:

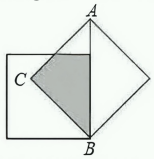


Solution 1 :

Text: Let the common vertex be **B**. Let **A** be the other endpoint of the diagonal of the right square that lies on the boundary of the left square, and let **C** be the adjacent vertex of the right square so that ∠ ACB=90 °. and BA is a side of the left square while the diagonal of the right square lies along the segment through B and A.",

Visual Gating: $\mathcal{I}$ = 0.51 $\mathcal{F}$ = 0.014 ⟶ **Keep Generation**

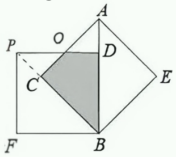
Image:



Text: Let the left square be **BDPF** with side length 1 and the right square be **ACBE** with side length 1. AC and PD intersect at point O. The diagonal of a unit square has length $\sqrt{2}$. To find the overlap, **draw CP** and consider the intersection region.

Visual Gating: $\mathcal{I}$ = 0.62 $\mathcal{F}$ = 0.021 ⟶ **Keep Generation**

Image:



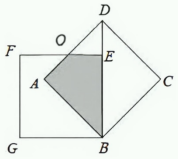
Text: The overlap region is the right triangle △PDB of the left square minus the small right triangle △PCO cut off by the diagonal alignment. Therefore, $S=S_{\triangle PDB}-S_{\triangle PCO}$ $=\frac{1}{2}\times1\times1-\frac{1}{2}\times(\sqrt{2}-1)^2=\sqrt{2}-1$. Therefore, **the option is A.** ✅

Solution 2:

Text: Let the common vertex be **B**, and let the right square be **ABCD** with AB side and BD its diagonal. The other square is **BEFG** with BE a side along BD of the first square. The overlap region is the quadrilateral **ABEO** inside the left square.

Visual Gating: $\mathcal{I}$ = 0.41 $\mathcal{F}$ = 0.016 ⟶ **Keep Generation**

Image:

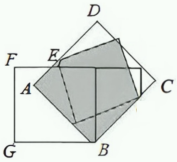


Text: The side length of each square is 1, so the diagonal BD has length $\sqrt{2}$.

Visual Gating: $\mathcal{I}$ = 0.22 $\mathcal{F}$ = 0.006 ⟶ **Abort Generation**

(Ours) Text: The overlap area is $S_{\triangle ABD}-S_{\triangle DOE}=\frac{1}{2}-\frac{1}{2}(\sqrt{2}-1)^2=\sqrt{2}-1$. Therefore, **the area of the overlap is A.** ✅

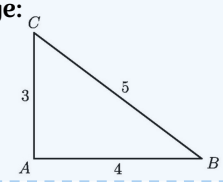
(MathCanvas) Image:



Text: The area of the overlap is close to $\frac{\sqrt{2}+1}{2}$. Hence, **the correct option is C.** ❌

Figure 14: Case study between our method and MathCanvas.

Question:
In the figure below, choose point D on BC so that △ACD and △ABD have equal perimeters. What is the area of △ ABD?
Options: A. $\frac{3}{4}$, B. $\frac{3}{2}$, C. 2, D. $\frac{12}{5}$, E. $\frac{5}{2}$

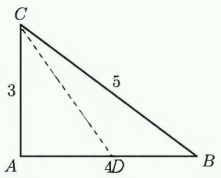
Input Image:


Solution:

Text: Let BD = x, so CD = BC - BD = 5 - x. The perimeters are: P_{ABD} = AB + AD + BD = 4 + AD + x. P_{ACD} = AC + AD + CD = 3 + AD + (5 - x) = 8 + AD - x. Equal perimeters give 4 + AD + x = 8 + AD - x, hence x = 2.

Visual Gating: $\mathcal{I}$ = 0.42 $\mathcal{F}$ = 0.007 ⟶ **Abort Generation**

(Ours) Text: The area of △ABC is $\frac{1}{2}AB \cdot AC = 6$. Also, the area of △ABD equals $\frac{BD}{BC}$ times △ABC. So the area of △ABD is $\frac{2}{5} \cdot 6 = \frac{12}{5}$. Therefore, **the correct option is D.** ✅

(MathCanvas) Image:


Text: Since $\triangle ACD$ and $\triangle ABD$ have equal perimeters and share side AD, the remaining sides satisfy $AC = AB$. So $\triangle ABC$ is isosceles with base AB.
Drop $AD \perp BC$ at D. Then $BD = DC = \frac{BC}{2}$.
From the diagram, $BC = AB = AC = 4$, and $AC = 3$ is given, giving $BD = 2$. By the Pythagorean theorem:

$$AB^2 = AD^2 + BD^2 \implies 16 = AD^2 + 4 \implies AD = 2\sqrt{3}.$$

Therefore:

$$S_{\triangle ABD} = \frac{1}{2} \cdot AB \cdot AD = \frac{1}{2} \cdot 4 \cdot 2\sqrt{3} = 4\sqrt{3}.$$

Since the options are unitless, $4\sqrt{3}$ corresponds to **option A.** ❌

Figure 15: Case study between our method and MathCanvas.

A.6 Additional Random-Drop Experiments

We further evaluate random visual-step dropping on BAGEL-Canvas at abort rates of 10%, 15%, 20%, 25%, and 50%. The results are averaged over three random masks and reported in Table 7. Although random dropping occasionally improves accuracy, increasing the abort rate does not consistently yield better performance, indicating that indiscriminate dropping may also discard beneficial visual generations.

To further isolate the effect of selective gating, we compare AdaViG with random dropping at matched abort rates of 29% on MathVista and 83% on DynaMath. AdaViG outperforms random dropping by 4.1% and 5.2% respectively. These results demonstrate that AdaViG's improvements stem from selectively aborting visual generations rather than merely reducing their frequency.

A.7 Additional Text-Only Experiments

To further investigate whether the improvements of ***AdaViG*** arise from selective visual gating rather than simply disabling visual generation, we evaluate a stronger text-only baseline on Bagel-Canvas. Unlike the original text-only baseline, which aborts every triggered visual generation, the stronger

Table 7: **Comparison with random visual-step dropping.** AdaViG consistently outperforms random dropping at matched abort rates.

	MathVista		DynaMath	
Method	**Drop (%)**	**Acc. (%)**	**Drop (%)**	**Acc. (%)**
Bagel-Canvas	0	77.9	0	42.3
Random	10	77.7	10	42.9
	15	77.8	15	43.3
	20	77.6	20	43.8
	25	77.8	25	43.6
	50	77.1	50	43.1
Random (matched)	29	77.2	83	42.8
AdaViG	29	**81.3**	83	**48.0**

Table 8: **Comparison with text-only reasoning baselines.** Accuracy (%) on Bagel-Canvas.

Method	**MathVista**	**WeMath**	**DynaMath**
Text-only (weak)	76.9	34.1	42.7
Text-only (strong)	76.9	34.3	42.8
AdaViG	**81.3**	**38.8**	**48.0**

baseline explicitly requires textual reasoning through prompting and prevents the model from emitting image-start tokens via constrained decoding.

As shown in Table 8, the two text-only baselines achieve nearly identical accuracy, suggesting that planning a subsequently aborted visual step has little impact on reasoning performance. In contrast, ***AdaViG*** outperforms the stronger text-only baseline by 4.4%–5.2%. These results further support the benefit of selectively retaining useful visual generations over entirely disabling visual reasoning.

A.8 More Ablation Studies on Bagel-Zebra-CoT

As shown in Tables 9 and 10, ***AdaViG*** consistently achieves the highest accuracy across all three benchmarks. Combining Generation Intent and Visual Fidelity outperforms using either signal alone, while the alternative gating combination leads to degraded performance. Moreover, ***AdaViG*** outperforms random aborting, text-only reasoning, and DAP-ICoT, demonstrating the benefit of selectively suppressing harmful visual generations.

A.9 Effect of the Thresholds $\tau_{\mathcal{I}}$ and $\tau_{\mathcal{F}}$

As shown in Tables 11 and 12, we further investigate the sensitivity of ***AdaViG*** to the gating thresholds $\tau_{\mathcal{I}}$ and $\tau_{\mathcal{F}}$. The thresholds are calibrated once per model on the MathVision testmini set and kept fixed across all evaluation benchmarks, without dataset-specific tuning.

Table 9: Performance of Bagel-Zebra-CoT under gating methods with different ($\mathcal{I}$,$\mathcal{F}$) combinations.

Method	Avg. Drop	Performance		
		MathVista	WeMath	DynaMath
Bagel-Zebra-CoT	–	81.0	34.5	39.0
w/ $\mathcal{I}$ only	64%	82.2	35.9	41.1
w/ $\mathcal{F}$ only	61%	81.9	36.1	41.6
w/ $\mathcal{I} \vee \mathcal{F}$	76%	78.2	31.5	38.4
w/ ***AdaViG***	57%	**83.4**	**38.1**	**43.5**

Table 10: Comparison with other image generation aborting strategies on Bagel-Zebra-CoT.

Method	Drop Rate	Performance		
		MathVista	WeMath	DynaMath
AdaViG	57%	**83.4**	**38.1**	**43.5**
Random	25%	78.2	34.3	38.9
	50%	80.4	34.3	37.7
Text-only	100%	77.1	32.5	37.5
DAP-ICoT [22]	16%	81.0	34.5	39.3

Table 11: Accuracy (%) on MathVista across $\tau_{\mathcal{I}} \in \{0.5, 0.6, 0.7\}$ with $\tau_{\mathcal{F}}$ fixed at 0.02.

Model	$\tau_{\mathcal{I}}$=0.5	$\tau_{\mathcal{I}}$=0.6	$\tau_{\mathcal{I}}$=0.7
Bagel-Canvas	80.6	**81.2**	81.0
Bagel-Zebra-CoT	**83.2**	82.3	82.1

Table 12: Accuracy (%) on MathVista across $\tau_{\mathcal{F}} \in \{0.01, 0.02, 0.04\}$ with $\tau_{\mathcal{I}}$ fixed at 0.6.

Model	$\tau_{\mathcal{F}}$=0.01	$\tau_{\mathcal{F}}$=0.02	$\tau_{\mathcal{F}}$=0.04
Bagel-Canvas	**81.1**	80.7	80.3
Bagel-Zebra-CoT	82.4	82.7	**83.1**

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: Our abstract and introduction accurately reflect the contributions of the paper, evidenced by the following sections.

   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 in Section 6.3.

   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: Our work does not include theoretical results.

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 have provided our code as the supplementary material.

   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: We have provided our code as the supplementary material.

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 have provided our code as the supplementary material, as well as experiment details and instructions. We also describe experiment details.

   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: [No]

   Justification: Our main results do not contain randomness as we fixed the seed.

   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 provide the details in Section 6.

   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: We have confirmed 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: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no 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 have made the citations.

    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 provide the implementation code in the supplementary materials.

    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: 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.
    - 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: We do not use LLMs for method development.

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.