TIMAGE: A GENERATIVE TEXT-IN-IMAGE PARADIGM FOR FINE-TUNING VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) often lose track of the right image regions during fine-grained spatial reasoning, because a textual query rarely carries any explicit geometric anchor into the pixel domain. Prevailing remedies either rewire the model's weights or pad the prompt with verbose instructions, yet neither reliably pins the language to the correct visual coordinates without eroding the backbone's general competence. We introduce *Timage*, a paradigm that recasts multimodal understanding as an alignment problem solved *at the input*: the query is drawn, as a typeset overlay, onto the image itself. The placement and appearance of this overlay are produced by a *Constrained Schrödinger Bridge* (cSB), an entropic optimal-transport sampler that factorizes layout synthesis into two coupled stochastic stages. The first stage, *Region Search*, transports noise toward query-aligned image zones while obeying a hard occlusion barrier that protects salient foreground content; the second stage, *Appearance Shaping*, sizes the glyphs through an "ink-budget" regularizer so that the rendered text stays legible and visually balanced. The resulting overlay behaves as an explicit attention beacon that channels the model's focus along spatial semantics. On the VMCBench and MME suite, Timage paired with a modest 7B backbone clearly overtakes far larger proprietary systems as well as parameter-tuned baselines. The study positions deliberate input reconstruction as a powerful, architecture-neutral lever for strengthening multimodal reasoning.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have reshaped how machines couple visual perception with linguistic reasoning, and they now span a wide task surface: answering questions about cluttered scenes, parsing documents, reading scientific charts, and more. Beneath these successes, however, lies a potential limitation of conventional multimodal interfaces: textual queries and visual inputs are initially represented in different modalities, leaving their spatial correspondence to be established implicitly by the model. A textual instruction (a question, a directive, or a reasoning cue) is tokenized into a discrete symbol stream, whereas the image enters as a dense, continuous tensor. The two interact later through multimodal fusion mechanisms such as cross-attention. Consequently, the model must associate abstract phrases "to the left of," or a positional reference inside a packed layout—with concrete visual regions. Such implicit correspondence can become challenging when tasks require fine spatial discrimination, dense text reading, or multi-step reasoning, motivating a more explicit mechanism for aligning query semantics with visual regions.

Existing attempts to improve multimodal adaptation broadly fall into two families, each with distinct trade-offs. The first, weight-space adaptation, includes parameter-efficient fine-tuning methods such as LoRA (Hu et al., 2022) and adapter modules (Chen et al., 2022; Sung et al., 2022; Zhang et al., 2021; Gao et al., 2024). These methods adapt internal parameters toward a downstream task distribution and can achieve strong task-specific performance, but require modifying or storing additional model parameters for adaptation. More importantly for our setting, they address multimodal adaptation through the model parameters rather than explicitly restructuring how query semantics are presented alongside the visual input. The second family, input-space prompting, represented by Visual Prompt Tuning, optimizes continuous learnable vectors appended to the input. While parameter-efficient, these vectors are not directly human-readable and do not explicitly encode natural-language semantics in the visual space. Consequently, they provide limited means to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: (a-c) Prior approaches stumble on feature-level decoupling, spatial misplacement, or visual occlusion. (d) Timage instead synthesizes query-conditioned semantic overlays via the cSB sampler. By binding the instruction to the image manifold at the input stage, Timage lifts complex spatial reasoning and returns better-grounded multimodal answers.

directly exploit the backbone's text-recognition capability through semantically meaningful visual cues.

These observations expose a missing piece in multimodal representation learning: there is no mechanism that explicitly turns the query text into a harmonious cross-modal cue co-located with the visual signal. We fill this gap with *Text-in-Image* (Timage), which reframes multimodal reasoning as input-level alignment on a shared manifold. Rather than feeding the query as an external token stream, Timage paints the natural-language instruction onto the image as a typeset overlay, producing a query-conditioned visual prompt (Figure 1(d)). As a paradigm, Timage offers three distinguishing properties. ❶ In contrast to the standard recipe of Figure 1(a), which embeds text and vision through separate branches, Timage **routes everything through a single visual input**. ❷ Timage **unlocks MLLMs' multimodal reasoning capabilities** via more complex text-in-image inputs, while keeping the underlying MLLM frozen during adaptation. ❸ Timage is **simultaneously human-legible and machine-actionable**: the cross-modal cue is literally visible, which is also a step toward interpretable multimodal AI.

Realizing this idea is not trivial. As Figure 1(b-c) shows, naively stamping the text as a fixed caption (Wang et al., 2025) or at a random location can be either too rigid or unstable. Moreover, valid overlay placement is inherently non-unique: multiple regions may simultaneously satisfy semantic relevance, readability, and non-occlusion requirements. We therefore formulate layout generation as distributional modeling rather than deterministic point prediction, instantiated through a *Constrained Schrödinger Bridge* (cSB). We decompose text-in-image rendering into two questions—*where* to write and *how* to write. For *where*, we encourage placement toward query-relevant content while enforcing a hard constraint that salient foreground objects remain unobstructed. For *how*, an ink usage budget adaptively controls glyph size for readable rendering. These objectives define control conditions that guide the bridge process toward feasible query-conditioned overlay layouts.

We evaluate Timage on VMCBench (Zhang et al., 2025) and MME (Fu et al., 2026). Built upon Qwen2.5-VL-7B, Timage achieves **91.3%**, **75.8%**, **98.5%**, and **92.6%** on General, Reasoning, OCR, and Doc&Chart, respectively, attaining the best performance across all four VMCBench categories. It also achieves the best or tied-best results on all four reported MME tasks. Despite using only a 7B backbone, Timage consistently surpasses substantially larger models, including GPT-4o and Qwen2-VL-72B, across all VMCBench categories. Following the weighted-average VMCBench protocol, Timage reaches **87.7%** accuracy, outperforming LoRA and full fine-tuning by **+3.2** and **+2.5** percentage points, respectively, suggesting that input reconstruction can be more effective than parameter adaptation for multimodal reasoning. Cross-architecture experiments further confirm the generality of Timage. Across LLaVA-1.5-7B, Qwen2-VL-7B, InternVL-2-8B, Qwen2.5-VL-7B, and Qwen3-8B, Timage consistently improves performance by **+4.8**–**+8.6** percentage points, with an average gain of **+6.1** points.

Our contributions are: ❶ We present Timage, a pioneering paradigm that fuses linguistic semantics with the visual signal by rendering the text query as an optimized overlay, generating the single modality as the input of MLLMs, which is plug-and-play for various MLLMs, and visibly interpretable. ❷ We cast overlay-layout synthesis as a Constrained Schrödinger Bridge, yielding a principled generative procedure whose energy functional trades off semantic pull against hard physical constraints. ❸ We set new state-of-the-art results on VMCBench and MME in a frozen MLLM setting, with stronger generalization and efficiency than both large proprietary models and parameter-efficient fine-tuning.

2 Related Work

Fine-Tuning of Vision-Language Models. Adapting Vision-Language Models has been widely studied through parameter-efficient strategies (Han et al., 2024; Ding et al., 2023; Xu et al., 2026). Representative approaches include adapter-based methods (Sung et al., 2022; Zhang et al., 2021; Gao et al., 2024; Hu et al., 2023; Chen et al., 2022), which insert lightweight task-specific modules into a largely frozen backbone; Low-Rank Adaptation and its variants (Hu et al., 2022; Liu et al., 2024; Agiza et al., 2024; Wang et al., 2024; Liang & Li, 2024; Zanella & Ben Ayed, 2024; Luo et al., 2023), which parameterize model updates with low-rank factors; and Visual Prompt Tuning and related methods (Han et al., 2023; Das et al., 2023; Huang et al., 2023; Sohn et al., 2023; Gao et al., 2022), which introduce learnable prompt tokens into the visual encoder. While effective and parameter-efficient, these approaches primarily adapt model parameters or latent representations, leaving the correspondence between linguistic instructions and image-space locations to be established implicitly by the model. In contrast, Timage operates directly in the input space by rendering semantic cues onto the image and explicitly modeling where they should be placed, providing a complementary route to semantic–spatial alignment without modifying the underlying VLM.

Generation in Service of Multimodal Understanding. Folding generative steps into the understanding pipeline has become a fertile direction. The literature ranges from textual Chain-of-Thought extensions (Wei et al., 2022; Feng et al., 2023; Zhang et al., 2024) to visual-generative reasoning (Tian et al., 2024; Chen et al., 2020; Van den Oord et al., 2016; Reed et al., 2017; Esser et al., 2021; Van Den Oord et al., 2017; Razavi et al., 2019; Yu et al., 2022; 2021), where models emit intermediate sketches, heatmaps, or masks to scaffold hard logic. Other lines exploit interactive generation (JonathanHo & Abbeel, 2020; Kodaira et al., 2025; Song et al., 2020a;b; Lu et al., 2022; Rombach et al., 2022; Luo et al., 2023; Shih et al., 2023; Du et al., 2020), producing iterative cues that sharpen spatial grounding and cross-modal alignment downstream. Most of this work, however, treats the reasoning trace as a side modality or a separate step kept apart from the original visual input. While that decoupling cleanly separates perception from reasoning, it can introduce a mismatch between the reasoning context and the primary scene. Motivated by this, we ask whether the reasoning prompt can be inscribed directly into the original image, yielding a self-contained, instruction-aware context.

3 Proposed Method

We propose Timage, a paradigm for visual-language interaction that writes the natural-language query into the image as a semantically aligned, spatially admissible overlay. Whereas conventional systems carry text as an external token sequence detached from image geometry, Timage produces a query-conditioned visual prompt through a controllable stochastic generator. This couples linguistic meaning with spatial context directly: the overlay nudges a downstream model toward task-relevant regions while a hard constraint forbids it from covering salient foreground content.

3.1 Overview and Notation

Take an image x_0 and a query q; our aim is to inscribe q onto x_0, much as one would letter a caption onto a painting. Two decisions govern the result: *where* the text lives and *how* it looks. We encode the first as a placement field and the second as a style field, and we score any candidate glyph layout $\mathcal{T}$ with a single *writing energy*

$$E(\mathcal{T}; x_0, q) = \lambda_{\text{zon}}\, E_{\text{zon}}\big(\alpha(\mathcal{T}); x_0, q\big) + \lambda_{\text{sty}}\, E_{\text{sty}}\big(\beta(\mathcal{T})\big), \tag{1}$$

where $\alpha \in [0,1]^{H\times W}$ is the placement field marking where $\mathcal{T}$ is written, and β summarizes its appearance. The two terms are defined as follows (Section 3.2 makes E_zon and E_sty precise):

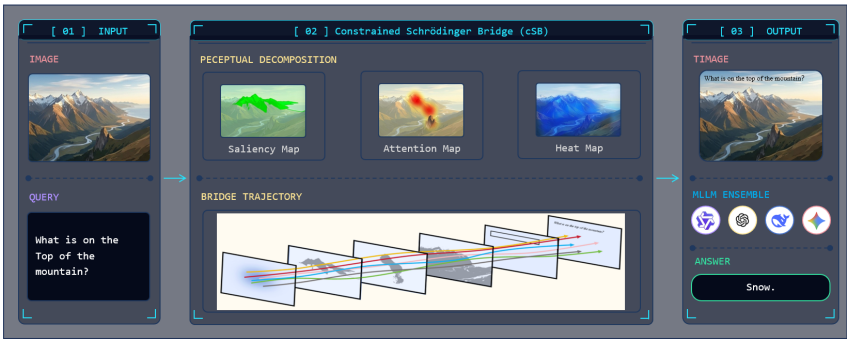

Figure 2: Overview of Timage. From an image and its question we build a feasible manifold Ω through an Admissible Mask that merges semantic relevance with a hard non-occlusion prior. A Constrained Schrödinger Bridge then renders the inlaid text via a projected stochastic process, enforcing geometric validity while improving downstream accuracy.

- *Placement energy* E_{zon}: rewards regions α that are query-relevant yet do not sit on top of the image content the query refers to.
- *Style energy* E_{sty}: imposes appearance priors–an ink budget and spatial smoothness–so that $\mathcal{T}$ stays readable. Note that β inherits constraints from the chosen placement α.

3.2 Building the Energy Function

Given the pair (x_0, q), we synthesize $x_1 = x_0 \oplus \mathcal{T}$, where $\mathcal{T}$ renders the content of q. Writing $\mathcal{T}$ requires a placement field $\alpha(\mathcal{T})$ and an appearance field $\beta(\mathcal{T})$ subject to the following constraints.

Placement Constraint E_{zon}. We want α (shorthand for $\alpha(\mathcal{T})$) to be *near* the image evidence implicated by q but *off* of it. Proximity helps the MLLM associate the inscribed $\mathcal{T}$ with the relevant region; non-coverage protects the very content the query depends on, especially query-related foreground objects. We make this concrete in two parts.

❶ *Semantic pull (soft).* To express soft alignment, we first form a query-relevance heatmap H_q. With $e_q = \text{Encoder}_{\text{text}}(q)$, we read cross-attention between e_q and the visual patches of a frozen VLM (e.g., Qwen2.5-VL) to produce H_q. The placement α should overlap with the spatial focus indicated by e_q, giving

$$E_{\text{sem}}(\alpha) = -\int \alpha(u)\, H_q(u)\, du, \tag{2}$$

with u a spatial coordinate; stronger overlap means lower energy.

❷ *Non-occlusion (hard).* We deliberately treat occlusion as a barrier rather than a soft penalty. We assemble an *Admissible Mask* $M_{\text{cand}} \in \{0,1\}^{H\times W}$ that delimits the feasible region by fusing object and saliency evidence:

$$M_{\text{cand}}(u) = \sigma\left(\mathbf{I} - \left(\bigcup_i O_i(u)\right) - \gamma\, S(u)\right), \tag{3}$$

where $\mathbf{I}$ is the all-ones $W \times H$ matrix, $O_i(u)$ is the instance mask of object i from a segmenter (e.g., Mask R-CNN), $S(u)$ is a dense saliency map, γ weights the saliency term, and σ binarizes via thresholding. Non-occlusion is enforced *hard*: we require $\alpha \subset M_{\text{cand}}$, so feasibility is guaranteed rather than merely encouraged.

Style Constraint E_{sty}. With α fixed, we render $\mathcal{T}$ strictly inside it and shape its appearance through β. In practice the dominant knob is font size, which we govern with a *Query-Adaptive Ink Budget*: just as physical writing consumes ink, we cap the rendered ink so glyph size adapts to the text and the scene. The target ink volume is

$$\text{ink}(\mathcal{T}) = \kappa \cdot \text{Length}(q) \cdot \text{Scale}(\alpha), \tag{4}$$

where Length counts tokens, Scale is the resolution factor of the placement region, and κ is the nominal ink per character per unit area.

3.3 CONSTRAINED SCHRÖDINGER BRIDGE

As with lettering a painting, neither the region nor the style is uniquely determined: many placements and many sizes are acceptable. We therefore want a method that *samples* from the set of admissible layouts rather than collapsing onto one. This is exactly what motivates us to synthesize the overlay through a *Constrained Schrödinger Bridge* (cSB): an optimal stochastic transport from a noise prior $\mu_0 = \delta$ to the inscribed text $\mathcal{T}$. We provide the formal derivation and probabilistic interpretation in Appendix.A.

Constrained Dynamics. We first transport the *placement field* α, keeping the trajectory inside the feasible manifold M_{cand} of Section 3.2. Unlike unconstrained SB, our energy target rewards only semantic alignment (Eq. (2)); spatial feasibility is delegated to the membership constraint $\alpha \subset M_{\text{cand}}$ from Eq. (3).

Following Song et al. (2020b), generation obeys a drift-diffusion SDE driven by a time-dependent velocity field v_θ:

$$d\mu_t = v_\theta(t, \mu_t, \mathbf{c})\, dt + \sqrt{2\mathcal{F}(t)}\, dW_t, \quad t \in [0, 1], \tag{5}$$

where $\mathcal{F}(t)$ is a fixed noise schedule and $\mathbf{c}$ bundles the conditions of Section 3.2. Concretely $\mathbf{c} = \{x_0, q, H_q, M_{\text{cand}}\}$ gathers the original input, the query-semantic cue, and the admissible region; it acts as the control signal throughout generation.

To merge these cues, at each step the network forms the composite input $[\mu_t; \Phi(\phi(t)); \Psi(\mathbf{c})]$, where $\phi(t)$ are Fourier time features broadcast to spatial size by Φ, and $\Psi(\mathbf{c})$ injects the semantic features (e_q, H_q) through cross-attention with $e_q = \text{Encoder}_{\text{text}}(q)$. To keep the path inside M_{cand}, we discretize with a *Projected Euler–Maruyama* scheme (Pierret, 2016); each step of size Δt reads

$$\mu_{t+\Delta t} = \mathcal{P}_\Omega\Big(\mu_t + v_t\, \Delta t + \sqrt{2\mathcal{F}(t)}\, \Delta W_t\Big), \tag{6}$$

where $v_t = v_\theta(t, \mu_t, \mathbf{c})$ is the predicted instantaneous drift and $\mathcal{P}_\Omega$ projects back onto the feasible set $\Omega = M_{\text{cand}}$.

Equation (6) cleanly separates two responsibilities: the conditioned drift v_t learns the *semantic direction*, while the projection $\mathcal{P}_\Omega$ enforces *spatial feasibility*. Running this to completion yields μ_1, a layout that satisfies both the semantic objective and the non-occlusion barrier; we set the placement field $\alpha = \mu_1$.

Next we generate the *appearance field* of $\mathcal{T}$. During this stage the ink budget enters the style energy E_{sty} as a regularizer; tuning λ_{sty} penalizes deviation so that the activated pixel count $\|\mathcal{T}\|_1$ approaches $\text{ink}(\mathcal{T})$. The glyph size thus adapts automatically–long queries draw a larger ink allotment (and may wrap to multiple lines), short queries stay compact–so the text settles naturally into the available space without crowding. Given region and size, line breaking follows automatically. Font and color are likewise carried through a Projected Euler–Maruyama solver: each iteration computes an intermediate noisy state seeded from default font and color values, which the drift then refines.

Training Objective. We learn v_θ with a two-part loss, $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cSB}} + \lambda\, \mathcal{L}_{\text{task}}$, jointly fitting the constraint energy and the downstream task.

Constrained flow-matching term ($\mathcal{L}_{cSB}$). Assuming a straight interpolation between μ_0 and μ_1, this term regresses the velocity onto the constant transport vector

$$\mathcal{L}_{\text{cSB}} = \mathbb{E}_t\Big[\big\|v_\theta(t, \mu_t, \mathbf{c}) - (\alpha^* - \mu_0)\big\|_2^2\Big], \tag{7}$$

where α^* is a pseudo-target for the placement field. Since the placement has no unique optimum, we obtain α^* by *approximate sampling* from the energy E_{zon} to build an empirical distribution $\tilde{\mu}_1$; we use Langevin dynamics (Gillespie, 2000) for this, with details in the supplement.

Task-guided term ($\mathcal{L}_{\text{task}}$). Given an image-query pair (x_0, q), the cSB generator produces the composite image $x_1 = x_0 \oplus \mathcal{T}(q)$. We then use the frozen MLLM $\mathcal{M}$ to evaluate the generated visual input. Importantly, after the composite image is constructed, the original textual query q is not separately provided to $\mathcal{M}$. We minimize

$$\mathcal{L}_{\text{task}} = -\log P_{\mathcal{M}}\left(y_{\text{gt}} \mid x_1\right). \tag{8}$$

Table 1: Performance comparison with state-of-the-art methods on VMCBench and MME. Results are reported as mean ± standard deviation over three independent runs with different random seeds.

	VMCBench				MME			
Model	**General**	**Reasoning**	**OCR**	**Doc&Chart**	**Exist**	**Count**	**Position**	**Commonsense**
Proprietary / Large-Scale Models (>40B)								
Qwen2-VL-72B	88.5 ± 2.1	72.6 ± 2.3	96.8 ± 1.8	90.1 ± 1.6	187.8 ± 7.5	168.3 ± 6.8	167.8 ± 4.3	152.6 ± 5.7
GPT-4o	85.2 ± 1.7	66.9 ± 1.9	96.4 ± 2.3	83.1 ± 2.1	192.2 ± 3.1	172.2 ± 3.5	173.9 ± 2.2	162.1 ± 4.7
Claude-3.5-Sonnet	81.3 ± 1.6	62.8 ± 2.7	93.4 ± 2.8	84.6 ± 3.6	186.1 ± 4.3	161.1 ± 7.4	163.3 ± 6.5	158.1 ± 5.9
Molmo-72B	82.9 ± 2.5	66.6 ± 2.6	94.7 ± 3.5	81.1 ± 3.7	185.0 ± 4.8	168.3 ± 4.3	166.1 ± 3.6	160.5 ± 3.2
Open-Source Mid-Scale Models (10B–40B)								
Qwen2.5-VL-14B	89.8 ± 2.6	72.6 ± 2.9	96.5 ± 3.3	89.9 ± 3.2	190.6 ± 5.3	170.6 ± 5.7	178.3 ± 5.5	150.5 ± 6.7
Cambrian-34B	83.7 ± 3.9	65.9 ± 3.2	95.7 ± 2.7	73.3 ± 2.6	183.3 ± 6.7	166.7 ± 4.5	154.4 ± 4.6	143.1 ± 5.3
VILA1.5-40B	82.5 ± 2.3	65.3 ± 2.1	93.2 ± 2.3	67.4 ± 2.9	184.4 ± 3.2	166.1 ± 5.2	160.0 ± 4.9	146.2 ± 5.6
CogVLM2-19B	78.1 ± 2.7	55.6 ± 3.9	92.3 ± 3.2	72.6 ± 3.7	190.0 ± 6.7	165.0 ± 7.3	153.3 ± 6.9	142.1 ± 6.0
Efficient Models (≤8B) & Our Method								
Qwen3-8B	87.3 ± 2.6	71.8 ± 2.7	96.4 ± 1.8	89.5 ± 2.3	194.4 ± 4.3	180.0 ± 4.1	176.7 ± 4.7	161.9 ± 5.3
Qwen2.5-VL-7B (Base)	85.6 ± 2.3	68.2 ± 2.0	95.3 ± 2.7	86.5 ± 2.1	182.2 ± 5.7	169.4 ± 5.0	173.9 ± 6.2	154.3 ± 6.9
Qwen2-VL-7B	84.5 ± 2.9	62.7 ± 3.2	96.4 ± 2.7	80.1 ± 3.7	179.4 ± 5.5	160.0 ± 5.2	174.4 ± 4.9	148.1 ± 5.3
Molmo-7B-D	73.2 ± 3.3	55.5 ± 2.8	91.7 ± 2.9	72.1 ± 3.6	172.8 ± 5.7	159.4 ± 6.7	142.2 ± 5.4	144.8 ± 6.1
Cambrian-8B	77.9 ± 2.1	56.4 ± 2.5	91.0 ± 1.9	65.4 ± 1.7	176.1 ± 4.9	163.3 ± 2.3	146.7 ± 2.9	124.3 ± 3.9
Phi-3-Vision	74.1 ± 2.9	56.4 ± 3.4	90.6 ± 3.7	73.8 ± 2.9	173.9 ± 4.2	160.0 ± 5.6	140.0 ± 6.2	132.9 ± 5.8
LLaVA1.5-7B	63.6 ± 3.9	44.7 ± 2.3	74.0 ± 1.8	35.0 ± 2.0	167.8 ± 6.2	153.3 ± 7.1	138.3 ± 6.9	126.9 ± 5.8
Timage (Base: Qwen2.5-VL-7B, K=1)	$\mathbf{90.1 \pm 1.1}$	$\mathbf{74.1 \pm 1.6}$	$\mathbf{97.8 \pm 1.7}$	$\mathbf{91.0 \pm 2.0}$	$\mathbf{195.2 \pm 2.2}$	$\mathbf{176.9 \pm 1.4}$	$\mathbf{179.5 \pm 1.3}$	$\mathbf{169.6 \pm 2.3}$
Timage (Base: Qwen2.5-VL-7B, K=5)	$\mathbf{91.3 \pm 0.9}$	$\mathbf{75.8 \pm 1.1}$	$\mathbf{98.5 \pm 1.2}$	$\mathbf{92.6 \pm 0.8}$	$\mathbf{198.3 \pm 2.4}$	$\mathbf{180.0 \pm 1.9}$	$\mathbf{181.7 \pm 1.7}$	$\mathbf{170.0 \pm 1.9}$
Base: Qwen2.5-VL-7B, K=5	*(+5.7)*	*(+7.6)*	*(+3.2)*	*(+6.1)*	*(+16.1)*	*(+10.6)*	*(+7.8)*	*(+15.7)*

Remark on the hybrid loss. The two terms are complementary. The task-guided term is direct: it back-propagates the downstream signal to improve placement and style. The constraint term is indispensable too: it supplies a placement prior that sharply shrinks the search space during diffusion and speeds convergence.

4 EXPERIMENTS

Datasets and Protocol We evaluate Timage on VMCBench and MME. VMCBench comprises **9,450** multiple-choice questions from 20 VQA datasets across four categories, while MME evaluates four perception and reasoning tasks. Timage samples K overlays per item and uses majority voting. The cSB module is trained on official training splits with the MLLM backbone frozen, and all results use official evaluation splits. We report VMCBench's weighted average accuracy for analyses.

4.1 COMPARISON WITH STATE-OF-THE-ART METHODS

Table 1 shows that **Timage** achieves consistently strong performance across both VMCBench and MME, despite using only a compact 7B backbone. On VMCBench, Timage attains the best score in all four categories, reaching **91.3%** on General, **75.8%** on Reasoning, **98.5%** on OCR, and **92.6%** on Doc&Chart. Compared with the Qwen2.5-VL-7B base model, Timage improves these categories by **5.7**, **7.6**, **3.2**, and **6.1** percentage points, respectively, with the largest gains appearing in Reasoning and Doc&Chart. Notably, it also surpasses substantially larger models such as Qwen2-VL-72B and GPT-4o across all four VMCBench categories. The advantage further extends to MME, where Timage achieves **198.3** on Existence, **180.0** on Count, **181.7** on Position, and **170.0** on Commonsense, yielding the best or tied-best result in every reported task. Relative to the base model, these correspond to gains of **+16.1**, **+10.6**, **+7.8**, and **+15.7** points, respectively. Overall, the consistent improvements across general perception, complex reasoning, OCR, document understanding, spatial perception, and commonsense reasoning demonstrate the effectiveness and robustness of Timage. These results suggest that embedding query-conditioned visual guidance into the image can strengthen multimodal reasoning without relying on increased model scale.

4.2 COMPARISON WITH ALTERNATIVE STRATEGIES

To probe Timage from multiple angles, we compare it against a broad spectrum of adaptation and prompting paradigms on the Qwen2.5-VL-7B backbone over VMCBench. As shown in Table 2, the baselines span five families: (1) *standard text prompting*, including direct prompting and Chain-of-Thought (CoT); (2) *heuristic and spatial visual prompting methods*, which rely on predefined placement or marking strategies; (3) *deterministic text-in-image prompting*, which optimizes prompt placement without updating the MLLM; (4) *existing visual prompting methods*, such as VPT and CVPT, which require few-shot task-specific training; and (5) *parameter-efficient or full fine-tuning*, which adapts model parameters using the full training set. In contrast to parameter-based adaptation methods, Timage keeps the MLLM backbone frozen and only trains the lightweight Constrained Schrödinger Bridge (cSB) to generate semantically informed text-in-image layouts. This design achieves the strongest performance across all four evaluation dimensions, reaching **91.3%**

Table 2: Comparison with representative adaptation and prompting paradigms on VMCBench.

Method	Category	Training	General	Reasoning	OCR	Doc&Chart
Standard Baselines						
Base Model (Text Only)	Text Prompt	No	85.6	68.2	95.3	86.5
+ Chain-of-Thought (CoT)	Text Prompt	No	86.1	69.5	95.8	87.2
Heuristic & Spatial Visual Prompts						
Random Text Overlay	Heuristic Prompt	No	86.9	69.5	95.8	86.1
Top-Left Text Placement	Heuristic Prompt	No	86.2	69.0	95.5	87.0
Saliency-Based Placement	Heuristic Prompt	No	86.1	68.5	96.0	88.2
Bounding Box Prompt	Spatial Marker	No	86.8	69.9	95.2	86.0
Deterministic Text-in-Image Baselines						
Semantic Greedy Placement	Text-in-Image	No	89.3	72.8	97.3	88.4
Direct Energy Minimization	Text-in-Image	No	89.7	72.1	96.2	88.5
Existing Visual Prompting						
VPT-shallow (Jia et al., 2022)	Visual Prompt	Yes (Few-shot)	86.8	70.8	96.1	88.4
VPT-deep (Jia et al., 2022)	Visual Prompt	Yes (Few-shot)	87.5	72.1	96.4	89.6
CVPT (Huang et al., 2025)	Visual Prompt	Yes (Few-shot)	88.4	73.5	96.9	90.8
Parameter-Efficient Fine-tuning						
Adapter (Chen et al., 2022)	PEFT	Yes (Full)	88.2	72.9	96.8	90.1
LoRA (Hu et al., 2022) (Rank=64)	PEFT	Yes (Full)	89.1	74.2	97.2	91.5
Full Fine-tuning	Fine-tuning	Yes (Full)	89.8	75.1	97.5	92.3
Timage (Ours)	**Text-in-Image**	**Yes (cSB only)**	**91.3**	**75.8**	**98.5**	**92.6**

on General, **75.8%** on Reasoning, **98.5%** on OCR, and **92.6%** on Doc&Chart, while keeping the underlying MLLM frozen.

Semantics beat heuristics. Does merely placing text on the image suffice? *Random Text Overlay* and *Top-Left Placement* provide only modest improvements over the text-only baseline across the four dimensions, while *Saliency-Based Placement* improves Doc&Chart to 88.2% but remains close to the baseline on General and Reasoning. *Bounding Box Prompt*, which provides spatial guidance without semantic text, similarly yields limited gains. In contrast, Timage reaches **91.3%**, **75.8%**, **98.5%**, and **92.6%** across the four dimensions. These results suggest that spatial placement alone is insufficient, while incorporating semantic information into the visual prompt provides a more effective mechanism for text-in-image prompting.

Versus visual prompting. We use *natural-language content* rather than learned continuous visual prompt representations. CVPT reaches 88.4% on General, 73.5% on Reasoning, 96.9% on OCR, and 90.8% on Doc&Chart, outperforming the VPT variants across most dimensions. Timage further improves upon CVPT by **2.9**, **2.3**, **1.6**, and **1.8** percentage points, respectively. Unlike these visual prompting approaches, Timage constructs semantically meaningful text-in-image layouts while keeping the underlying MLLM frozen. The consistent gains suggest that explicit semantic content can complement spatial adaptation when constructing visual prompts for multimodal reasoning.

Versus parameter tuning. Timage also compares favorably with parameter-efficient and full fine-tuning. Compared with LoRA, Timage improves General, Reasoning, OCR, and Doc&Chart by **2.2**, **1.6**, **1.3**, and **1.1** percentage points, respectively. Compared with *Full Fine-tuning*, the corresponding improvements are **1.5**, **0.7**, **1.0**, and **0.3** percentage points. These results indicate that, under this evaluation setting, reformulating the visual input through cSB provides an effective alternative to parameter adaptation, achieving consistent improvements without updating the underlying MLLM parameters.

Why cSB instead of simpler placement optimization? Semantic placement helps, but deterministic optimization remains less effective than cSB in our experiments. *Semantic Greedy Placement* and *Direct Energy Minimization* achieve 89.3% and 89.7% on General, respectively, yet both trail Timage (91.3%). The gap is more pronounced on Reasoning and Doc&Chart, where Timage reaches 75.8% and 92.6%, outperforming these deterministic alternatives by **3.0–3.7** and **4.1–4.2** percentage points, respectively. These results suggest that point-wise placement optimization may be less effective at jointly balancing semantic relevance and spatial feasibility, while cSB performs constrained distribution matching to generate more effective text-in-image layouts.

4.3 Ablation Study

Table 3 isolates each component. From a random-placement base, adding Semantic Pull (E_{sem}) lifts accuracy by +2.5%, confirming the value of query-relevant alignment. Activating the Admissible Mask (M_{cand}) gives the single largest gain (+3.0%), underscoring that avoiding occlusion is critical in dense scenes. Style Regularization (E_{sty}) adds legibility and physical validity (+1.4% combined). Finally, the Stochastic Ensemble ($K = 5$) captures layout multi-modality for a closing +1.7%, reaching 87.7%. Standard diffusion performs worse than the proposed cSB under the same constrained setting. One possible explanation is that Timage requires generating layouts conditioned jointly on textual semantics and visual content, whereas cSB explicitly formulates this process as transport between constrained distributions. In addition, cSB converges with fewer sampling steps in our experiments ($\sim$20 vs. $\sim$50).

4.4 Qualitative Analysis

Figure 3 illustrates how Timage enhances MLLM reasoning. Left: random question embedding covers irrelevant content and misleads the model to 'Book'. Right: Timage renders a semantically grounded, non-occluding overlay that preserves foreground saliency and guides attention to the window region, yielding the correct 'Cat'. The baseline's random inscription happens to cover key content and pulls attention toward an irrelevant background object. Timage instead lays down a semantically grounded, non-occluding overlay: by seating the query beside the target region it keeps foreground saliency intact and redirects the attention map to the correct object. This corroborates that an optimized text layout works as a potent visual prompt for accurate reasoning.

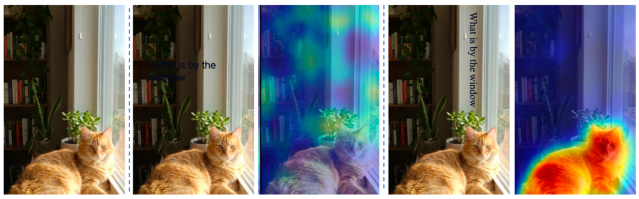

Figure 3: Qualitative comparison results.

Table 3: Incremental impact of Timage components on VMCBench. Overall accuracies discussed in the text are computed using the official VMCBench weighted-average protocol over its 20 constituent VQA datasets.

Configuration	**Sem.** (E_{sem})	**Mask** (M_{cand})	**Style** (E_{sty})	**Ens.** ($K = 5$)	**Avg.**
Random Placement (Base)	–	–	–	–	79.1
+ Semantic Pull	✓	–	–	–	81.6
+ Admissible Mask	✓	✓	–	–	84.6
+ Style Regularization	✓	✓	✓	–	86.0
Standard Diffusion + Constraints	–	–	–	–	83.2
Timage	✓	✓	✓	✓	**87.7**

Performance in complex scenarios. Baselines suffer from attentional drift in clutter, whereas Timage's spatial anchoring redirects focus to small, background targets (e.g., buses in Figure 4 top) without occlusion.

Reasoning enhancement under imperfect attention. Figure 4 bottom shows that the model maintains robust performance even when attention is not perfect for both original image and Timage (one person excluded in the attention map). We infer that the reason lies in that 1) the attached text guides the model to focus on regions outside the attention map, 2) the ***'reading' of the 'text-in-image' well has activated the model's potential for image understanding and reasoning***.

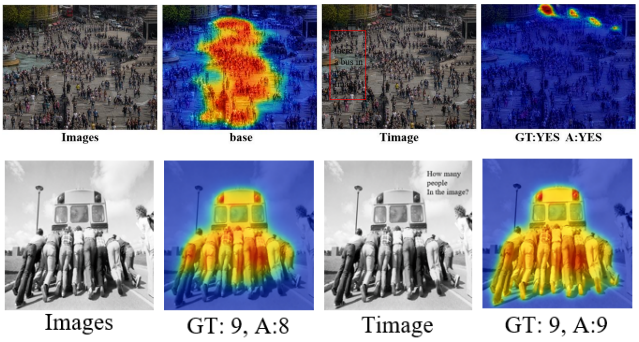

Figure 4: Qualitative comparison results in complex and challenging reasoning scenarios.

4.5 In-depth Analysis

Generalization on various MLLMs. To test universality, we attach Timage to five distinct backbones, from the classic LLaVA-1.5 to the recent Qwen3. Table 4 shows uniform improvement, averaging **+6.1%**. Even on the strong Qwen3-8B (87.3% base), Timage climbs to **92.1%** (**+4.8%**),

so the method does more than patch older models–it raises the ceiling of advanced VLMs. Gains concentrate on hard tasks, averaging **+6.8%** in Reasoning and **+5.6%** in Doc&Chart. That this holds across CLIP-based, hybrid-ViT, and dynamic-resolution designs points to a shared bottleneck in current VLMs: the disconnect between linguistic queries and visual geometry. By rebuilding the input to align intent with spatial context, Timage behaves as a robust, model-agnostic, plug-and-play module that can unlock latent capabilities across diverse frozen MLLMs.

Table 4: Cross-architecture generalization of Timage.

Backbone Model	Base Avg. Acc	Avg. ACC	Improvement (Δ)	Reasoning Δ	Doc&Chart Δ
LLaVA-1.5-7B	51.8	**57.4**	**+5.6**	+6.8	+5.2
Qwen2-VL-7B	78.1	**83.9**	**+5.8**	+7.1	+6.4
InternVL-2-8B	72.5	**78.3**	**+5.8**	+6.5	+5.9
Qwen2.5-VL-7B	79.1	**87.7**	**+8.6**	+7.6	+6.1
Qwen3-8B	87.3	**92.1**	**+4.8**	+5.9	+4.5
Average Improvement	–	–	**+6.1%**	**+6.8%**	**+5.6%**

OCR ability and text length. We test Timage on LLaVA-1.5-7B, which has relatively weak OCR capability. Timage still outperforms the baseline by 5.6 percentage points. These results suggest that Timage can remain effective even for MLLMs with relatively weaker OCR capability. In Figure 5, we show a case of long text.



Figure 5: Qualitative results for long text.

The adaptive ink constraint dynamically adjusts font and layout, ensuring the model perceives the content. Also, text prompts in VQA are typically not very long, and we leave the exploration of handling longer texts for future work.

Efficiency. We investigate the performance and efficiency under different K in Table 5. While accuracy improves by 0.4 percentage points as K increases from 1 to 5, the performance gains diminish for $K > 5$. To balance accuracy and computational overhead, we set $K = 5$ in our experiments.

Table 5: Performance and efficiency under different values of K.

Setting	Avg. Acc (%)	Latency (s)	GFLOPs	Setting	Avg. Acc (%)	Latency (s)	GFLOPs
$K = 1$	87.3	1.84	4.6	$K = 5$	**87.7**	**3.92**	**10.5**
$K = 3$	87.5	2.35	6.3	$K = 7$	88.0	5.48	14.7

To further gauge parameter efficiency, we compare Timage with LoRA and CVPT across trainable-parameter budgets. As Figure 6 shows, Timage attains higher VMCBench accuracy with far fewer trainable parameters. Even scaled to 40M parameters, LoRA only inches to 88.3, while Timage keeps its lead at minimal cost–exposing the diminishing returns of brute-force parameter growth and reinforcing the value of our constrained, geometry-aware adaptation.

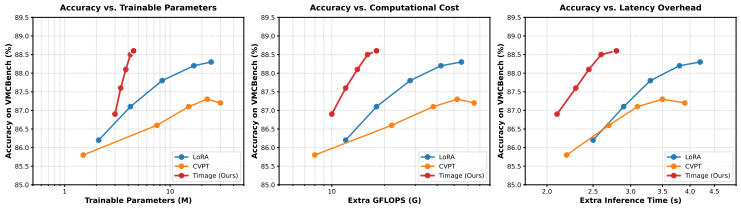

Figure 6: Efficiency-accuracy trade-offs on VMCBench.

More results. In the Appendix D and E, we provide comprehensive experiments about impact from font, color, text length, image resolution, etc on Timage, and qualitative results of failure cases.

5 Conclusion

We presented Timage, a text-in-image prompting framework that integrates linguistic instructions into visual inputs through semantically informed and spatially constrained layouts. By formulating layout generation as a Constrained Schrödinger Bridge, Timage adapts the visual input while keeping the underlying MLLM frozen. Experiments on VMCBench and MME demonstrate consistent improvements across different model backbones and adaptation baselines. Future work will extend Timage to multi-image and video settings.

AI USE STATEMENT

In this work, we used generative AI tools for code completion, unit test generation, and language editing. We have not used generative AI tools for formulating research hypotheses or drafting core scientific arguments. All generated code snippets were independently executed, unit-tested, and verified for correctness by the authors. We take full responsibility for the final content and technical artifacts presented in this work.

ETHICS STATEMENT

This work focuses on foundational algorithms and is evaluated entirely on standard, publicly available benchmark datasets. To the best of our knowledge, this work does not involve human participants, proprietary personal data, or direct applications with potential negative societal impacts. All scientific sources and baseline works have been properly cited.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide detailed descriptions of the methodological formulation, optimization procedure, and experimental configurations used in Timage. Specifically, the formulation of Text-in-Image generation, including the query-conditioned placement and style energies, the Admissible Mask for non-occlusion, and the Constrained Schrödinger Bridge (cSB) for stochastic overlay synthesis, is described in Section 3. We further detail the projected stochastic dynamics, flow-matching objective, task-guided optimization, and straight-through gradient estimation used to train the cSB module. Implementation details, including the optimizer, learning rate, weight decay, training epochs, batch size, ink-budget coefficient, software environment, and the self-consistency setting with $K = 5$, are provided in Section 3 and the appendix. We evaluate Timage on the publicly available VMCBench and MME benchmarks using their official evaluation splits and protocols, while keeping the underlying MLLM backbone frozen and training only the lightweight cSB module. Additional analyses and ablation studies on component design, cross-architecture generalization, computational efficiency, rendering configurations, and failure cases are provided in the appendix. The complete source code and configuration files will be released upon publication to facilitate reproducibility and further research.

REFERENCES

Ahmed Agiza, Marina Neseem, and Sherief Reda. Mtlora: Low-rank adaptation approach for efficient multi-task learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16196–16205, 2024. 3

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020. 3

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 1, 3, 7

Rajshekhar Das, Yonatan Dukler, Avinash Ravichandran, and Ashwin Swaminathan. Learning expressive prompting with residuals for vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3366–3377, 2023. 3

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 3

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020. 3

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021. 3

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:70757–70798, 2023. 3

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026. 2

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International journal of computer vision*, 132(2):581–595, 2024. 1, 3

Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N Metaxas. Visual prompt tuning for test-time domain adaptation. *arXiv preprint arXiv:2210.04831*, 2022. 3

Daniel T Gillespie. The chemical langevin equation. *The Journal of Chemical Physics*, 113(1): 297–306, 2000. 5

Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E^ 2vpt: An effective and efficient approach for visual prompt tuning. *arXiv preprint arXiv:2307.13770*, 2023. 3

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024. URL https://arxiv.org/abs/2403.14608. 3

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022. 1, 3, 7

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Conference on Empirical Methods in Natural Language Processing*, pp. 5254–5276, 2023. 3

Lingyun Huang, Jianxu Mao, Junfei Yi, Ziming Tao, and Yaonan Wang. Cvpt: Cross visual prompt tuning. In *IEEE/CVF International Conference on Computer Vision*, pp. 848–858, 2025. 7

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10878–10887, 2023. 3

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022. 7

AjayJain JonathanHo and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. 3

Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, Masayoshi Tomizuka, et al. Streamdiffusion: A pipeline-level solution for real-time interactive generation. In *IEEE/CVF International Conference on Computer Vision*, pp. 12371–12380, 2025. 3

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024. 3

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024. 3

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 3

Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick Von Platen, ApolinÃAario Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023. 3

Frédéric Pierret. A non-standard-euler–maruyama scheme. *Journal of Difference Equations and Applications*, 22(1):75–98, 2016. 5

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in Neural Information Processing Systems*, 32, 2019. 3

Scott Reed, Aäron Oord, Nal Kalchbrenner, Sergio Gómez Colmenarejo, Ziyu Wang, Yutian Chen, Dan Belov, and Nando Freitas. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, pp. 2912–2921, 2017. 3

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. 3

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:4263–4276, 2023. 3

Kihyuk Sohn, Huiwen Chang, José Lezama, Luisa Polania, Han Zhang, Yuan Hao, Irfan Essa, and Lu Jiang. Visual prompt tuning for generative transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19840–19851, 2023. 3

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a. 3

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b. 3, 5

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022. 1, 3

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 3

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in Neural Information Processing Systems*, 29, 2016. 3

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931, 2024. 3

Zhaochen Wang, Yiwei Wang, and Yujun Cai. Cure or poison? embedding instructions visually alters hallucination in vision-language models. *arXiv preprint arXiv:2508.01678*, 2025. 2

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. 3

Lingling Xu, Haoran Xie, S Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. 3

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 3

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 3

Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1593–1603, 2024. 3

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 1, 3

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024. 3

Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah Aklilu, Alejandro Lozano, Anjiang Wei, et al. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Conference*, pp. 29580–29590, 2025. 2

A WHY CONSTRAINED SCHRÖDINGER BRIDGE?

Overview. The placement of a text overlay is generally non-unique. For an image-query pair (x_0, q), multiple spatial regions may be simultaneously close to query-relevant visual evidence, free of foreground occlusion, and large enough for rendering the query. This motivates modeling text placement as a *distribution* over feasible layouts rather than as a single deterministic point estimate.

Our formulation consists of two coupled stages. First, an entropy-regularized terminal optimization defines a target distribution over semantically favorable and spatially admissible layouts. Second, a Schrödinger Bridge transports a noise prior to this target distribution while remaining close to a reference stochastic process. The projected flow-matching procedure used in Timage provides a tractable approximation to this constrained bridge.

A.1 FEASIBLE LAYOUT SPACE

Let $\alpha \in \mathcal{X}$ denote the placement field of the text overlay. Given an image-query pair (x_0, q), let $M_{\text{cand}}(x_0, q)$ be the admissible mask constructed from foreground segmentation and saliency evidence. We define the feasible layout space as

$$\Omega(x_0, q) = \{\alpha \in \mathcal{X} : \alpha(u) \le M_{\text{cand}}(u),\ \forall u\}\,. \tag{9}$$

The semantic placement energy introduced in the main paper is

$$E_{\text{zon}}(\alpha; x_0, q) = -\int \alpha(u) H_q(u)\, du, \tag{10}$$

where H_q is the query-conditioned relevance map derived from the frozen vision-language model.

A low value of E_{zon} corresponds to a layout that is close to regions indicated by the query-conditioned semantic evidence, while membership in Ω enforces the estimated non-occlusion constraint.

A.2 ENTROPY-REGULARIZED TERMINAL DISTRIBUTION

Let $r_1(\alpha \mid x_0, q)$ denote a reference distribution over terminal layouts. This reference distribution may be induced by a simple spatial prior or by the terminal marginal of a reference diffusion process. We define the desired terminal distribution as the solution of

$$\begin{aligned} \nu^\star = \arg\min_{\nu \in \mathcal{P}(\Omega)} \quad & \mathbb{E}_{\alpha\sim\nu}\left[E_{\text{zon}}(\alpha; x_0, q)\right] \\ & + \tau\, \mathrm{KL}\left(\nu \| r_1(\cdot \mid x_0, q)\right), \end{aligned} \tag{11}$$

where $\mathcal{P}(\Omega)$ denotes the set of probability measures supported on Ω, and $\tau > 0$ controls the entropy–energy trade-off.

Under standard absolute-continuity assumptions, the minimizer of Eq. (11) has the Gibbs form

$$\nu^\star(\alpha \mid x_0, q) = \frac{\mathbf{1}_\Omega(\alpha)\, r_1(\alpha \mid x_0, q) \exp\!\left(-\frac{E_{\text{zon}}(\alpha; x_0, q)}{\tau}\right)}{Z(x_0, q)}, \tag{12}$$

where

$$Z(x_0, q) = \int_\Omega r_1(a \mid x_0, q) \exp\!\left(-\frac{E_{\text{zon}}(a; x_0, q)}{\tau}\right) da \tag{13}$$

is the normalization constant.

Thus, feasible layouts with lower semantic energy receive larger probability mass, while multiple near-optimal layouts can coexist. Importantly, $\nu^\star$ is a *distribution* rather than a single preferred location.

A.3 Schrödinger Bridge Between Noise and the Target Layout Distribution

We next define the bridge process. Let $\mathbb{P}$ be a reference path measure induced by the diffusion

$$d\alpha_t = \sqrt{2\mathcal{F}(t)}\, dW_t, \qquad \alpha_0 \sim \mu_0, \tag{14}$$

where μ_0 is the initial noise distribution and $\mathcal{F}(t)$ is a prescribed diffusion schedule.

The constrained Schrödinger Bridge is then defined as the path measure

$$\begin{aligned} \mathbb{Q}^\star = \arg\min_{\mathbb{Q}} \quad & \mathrm{KL}(\mathbb{Q}\|\mathbb{P}) \\ \text{s.t.} \quad & \mathbb{Q}_0 = \mu_0, \qquad \mathbb{Q}_1 = \nu^\star, \end{aligned} \tag{15}$$

where the support of $\nu^\star$ is restricted to Ω by Eq. (12).

This formulation gives a precise interpretation of cSB. The terminal distribution is not chosen as a single deterministic optimum; instead, the bridge transports the initial noise distribution to an entropy-regularized distribution of admissible layouts while minimizing the deviation from the reference stochastic process.

Equivalently, the constraint can be incorporated directly into the terminal measure through

$$\mathrm{supp}(\mathbb{Q}_1) \subseteq \Omega(x_0, q). \tag{16}$$

A.4 Controlled Diffusion Representation

The Schrödinger Bridge in Eq. (15) admits a controlled stochastic-process interpretation. In particular, the optimal path measure can be represented by a controlled diffusion of the form

$$d\alpha_t = u_t(\alpha_t, c)\, dt + \sqrt{2\mathcal{F}(t)}\, dW_t, \tag{17}$$

where $c = \{x_0, q, H_q, M_{\text{cand}}\}$ denotes the conditioning variables and u_t is the bridge drift.

For the reference process in Eq. (14), the path-space KL divergence is equivalent to a quadratic control cost under the usual Girsanov regularity conditions:

$$\mathrm{KL}(\mathbb{Q}\|\mathbb{P}) = \frac{1}{4}\mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \frac{\|u_t(\alpha_t, c)\|_2^2}{\mathcal{F}(t)}\, dt\right], \tag{18}$$

up to the common endpoint terms determined by the prescribed marginals. Therefore, the bridge seeks the least-distorting stochastic control that transports μ_0 to $\nu^\star$.

In implementation, we enforce feasibility using the projected update

$$\alpha_{t+\Delta t} = \Pi_\Omega\left(\alpha_t + v_\theta(t, \alpha_t, c)\Delta t + \sqrt{2\mathcal{F}(t)}\, \Delta W_t\right), \tag{19}$$

where

$$\Pi_\Omega(z) = \arg\min_{\alpha \in \Omega} \|\alpha - z\|_2^2. \tag{20}$$

The learned field v_θ approximates the bridge drift, whereas Π_Ω serves as a discrete feasibility operator that prevents the generated placement from leaving the estimated admissible region.

We emphasize that Eq. (19) is a practical projected discretization of the constrained bridge dynamics. It is not claimed to be an exact closed-form discretization of the continuous-time Schrödinger system. This distinction separates the theoretical bridge formulation from its tractable numerical realization.

A.5 Flow-Matching Approximation

Direct estimation of the optimal bridge drift can be computationally demanding. We therefore use conditional flow matching to learn a neural approximation of the bridge drift.

Let

$$\alpha_1 \sim \nu^\star \tag{21}$$

be a sample from the target terminal distribution, and define the interpolating path

$$\alpha_t = (1-t)\alpha_0 + t\alpha_1, \qquad t \in [0,1]. \tag{22}$$

The conditional target velocity is

$$u_t^{\text{target}} = \frac{d\alpha_t}{dt} = \alpha_1 - \alpha_0. \tag{23}$$

The corresponding conditional flow-matching objective is

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\left[\|v_\theta(t,\alpha_t,c) - (\alpha_1-\alpha_0)\|_2^2\right]. \tag{24}$$

The minimizer of Eq. (24) is the conditional mean velocity

$$v^\star(t,\alpha,c) = \mathbb{E}\left[\alpha_1 - \alpha_0 \mid \alpha_t = \alpha, c\right]. \tag{25}$$

Therefore, the stochasticity of cSB enters through sampling $\alpha_1 \sim \nu^\star$ rather than through arbitrary perturbation of a single deterministic target. In practice, the terminal samples used for training are approximated using the constrained energy-based sampling procedure described in the main paper.

A.6 Deterministic Placement as a Limiting Case

A deterministic placement method solves

$$\hat{\alpha}_{\text{det}} = \arg\min_{\alpha\in\Omega} E_{\text{zon}}(\alpha; x_0, q). \tag{26}$$

The relationship between this point-estimation formulation and cSB follows directly from Eq. (12). As $\tau \to 0$, the Gibbs distribution concentrates its mass on the global minimizers:

$$\lim_{\tau\to 0} \nu_\tau^\star \text{ is supported on } \arg\min_{\alpha\in\Omega} E_{\text{zon}}(\alpha; x_0, q). \tag{27}$$

If the minimizer is unique, then

$$\nu_\tau^\star \Rightarrow \delta_{\alpha^\star} \qquad \text{as } \tau \to 0. \tag{28}$$

Hence, deterministic energy minimization is a zero-temperature, single-mode limit of the distributional formulation. For finite τ, however, multiple feasible near-optimal layouts can retain non-zero probability mass.

This distinction is important for text placement because there may be several semantically equivalent locations. For example, a query may be correctly associated with the same visual object by placing the overlay slightly above, below, or beside that object, provided that the text remains readable and does not occlude critical content.

Thus, the key distinction is not that cSB is universally better than every deterministic predictor. Rather,

$$\{\text{deterministic placements}\} \subset \{\text{probability distributions over feasible placements}\}, \tag{29}$$

with deterministic placement corresponding to a Dirac distribution.

The additional modeling capacity of cSB is therefore its ability to represent uncertainty and multimodality in feasible spatial hypotheses.

A.7 Task-Guided Refinement

The semantic terminal energy encourages the generated overlay to be spatially aligned with query-relevant visual evidence, but it does not directly measure whether the resulting composite image improves downstream MLLM prediction. We therefore use the frozen MLLM as a task-aware supervisor to further refine the learned transport.

It is important to distinguish the conditioning of the Timage generator from the input to the downstream MLLM. Given an image-query pair (x_0, q), the query q is used by the cSB generator to

determine the placement and appearance of the rendered overlay. After the composite image has been constructed, however, the original textual query q is *not* separately provided to the downstream MLLM. The frozen MLLM receives only the query-embedded visual input.

More precisely, for a generated overlay $\mathcal{T}(\alpha, \beta; q)$, we define the composite image as

$$x_1(\alpha, \beta; q) = x_0 \oplus \mathcal{T}(\alpha, \beta; q), \tag{30}$$

where q conditions the overlay-generation process but is not an additional input to the frozen MLLM $\mathcal{M}$. The downstream prediction therefore takes the form

$$y_{\text{pred}} = \mathcal{M}\left(x_1(\alpha, \beta; q)\right). \tag{31}$$

Accordingly, we define the task energy as

$$E_{\text{task}}(\alpha, \beta) = -\log P_{\mathcal{M}}\left(y_{\text{gt}} \mid x_1(\alpha, \beta; q)\right), \tag{32}$$

or, equivalently,

$$E_{\text{task}}(\alpha, \beta) = -\log P_{\mathcal{M}}\left(y_{\text{gt}} \mid x_0 \oplus \mathcal{T}(\alpha, \beta; q)\right). \tag{33}$$

This formulation makes the information flow explicit:

$$(x_0, q) \xrightarrow{G_\theta} x_1 \xrightarrow{\mathcal{M}} y_{\text{pred}}, \tag{34}$$

where G_θ denotes the cSB-based Timage generator. Thus, q conditions G_θ, whereas the downstream MLLM $\mathcal{M}$ operates only on the generated composite image x_1. In particular, there is no second language-channel copy of the sample-specific query provided to $\mathcal{M}$.

The complete terminal preference can conceptually be written as

$$E_{\text{total}} = \lambda_{\text{zon}} E_{\text{zon}} + \lambda_{\text{sty}} E_{\text{sty}} + \lambda_{\text{task}} E_{\text{task}}. \tag{35}$$

Under a fully energy-based formulation, this would induce the terminal measure

$$\nu^\star(\alpha, \beta) \propto \mathbf{1}_\Omega(\alpha)\, r_1(\alpha, \beta) \exp\left(-\frac{E_{\text{total}}(\alpha, \beta)}{\tau}\right). \tag{36}$$

In our implementation, we first construct placement pseudo-targets from the semantic and spatial constraints, and subsequently refine the learned transport using the downstream task loss. The corresponding practical objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cSB}} + \lambda_{\text{task}} \mathcal{L}_{\text{task}}, \tag{37}$$

with

$$\mathcal{L}_{\text{task}} = -\log P_{\mathcal{M}}\left(y_{\text{gt}} \mid x_1\right). \tag{38}$$

Since $\mathcal{M}$ is frozen, the task-guided signal does not update the MLLM parameters. Instead, gradients are propagated through the rendered composite image to refine the cSB generator, encouraging it to produce layouts and appearances that are not only semantically and spatially admissible but also beneficial for downstream prediction.

The task-guided term should therefore be interpreted as a practical refinement of the learned bridge rather than as a claim that the final hybrid training objective exactly solves the closed-form continuous-time Schrödinger system. Importantly, this refinement does not alter the single-input inference protocol: once the query has been embedded into the image by Timage, the downstream MLLM receives the resulting composite image alone.

A.8 Why the Distributional Formulation is Useful

The above derivation identifies three properties of cSB that are not captured by a point-wise placement optimizer.

(1) Multimodal spatial hypotheses. A deterministic predictor selects a single layout $\hat{\alpha}_{\text{det}}$. In contrast, cSB models $p(\alpha \mid x_0, q)$ and can preserve several feasible near-optimal layouts.

(2) Entropy-controlled exploration. The temperature τ determines the balance between semantic energy minimization and distributional diversity. As $\tau \to 0$, the model approaches deterministic optimization; for finite τ, it can retain multiple plausible solutions.

(3) Test-time spatial self-consistency. Multiple layouts can be sampled for the same image-query pair and evaluated through the frozen MLLM. This provides a principled form of spatial hypothesis sampling rather than arbitrary input perturbation.

Importantly, this analysis does not claim universal performance dominance over arbitrary deterministic learned placement networks. Instead, it establishes that cSB optimizes a richer distributional object and contains deterministic placement as a limiting case. Whether this additional distributional flexibility improves downstream multimodal reasoning is an empirical question, which we investigate through controlled ablations and sampling studies.

B Controlled Analysis of Distributional Spatial Transport

Appendix A provides a distributional interpretation of cSB, under which deterministic placement can be viewed as a restricted single-hypothesis case. In this section, we complement the theoretical analysis with controlled experiments that isolate the empirical contributions of learned placement, distributional modeling, spatial feasibility, and test-time spatial sampling.

All learned variants are evaluated under matched settings: they use the same frozen MLLM backbone, training data, semantic conditioning, rendering pipeline, admissible-region construction, downstream task supervision, and comparable optimization budgets. Table 6 summarizes the results. The reported Avg. follows the weighted VMCBench evaluation protocol and is therefore not the arithmetic mean of the four category scores.

(1) Learned deterministic placement vs. cSB. We first examine whether the improvement of Timage can be explained purely by learning a task-aware placement function. As shown in Table 6(A), Deterministic Learned Placement obtains an average accuracy of 85.8. Under the same spatial constraint and comparable learning conditions, single-sample cSB ($K = 1$) increases the average accuracy to 87.3, corresponding to a 1.5-point improvement.

The improvement is also consistent across all four evaluation categories. Deterministic Learned Placement achieves 88.5, 71.8, 96.4, and 88.0 on General, Reasoning, OCR, and Doc&Chart, respectively, whereas cSB ($K = 1$) improves these scores to 90.1, 74.1, 97.8, and 91.0. The largest category-level gain appears on Doc&Chart, with an improvement of 3.0 points, followed by a 2.3-point gain on Reasoning. This controlled comparison indicates that the benefit of cSB cannot be explained solely by replacing heuristic placement with a learned deterministic predictor.

The deterministic optimization baselines nevertheless remain competitive. Semantic Greedy Placement and Direct Energy Minimization obtain average accuracies of 86.8 and 86.5, respectively. However, both methods optimize a single layout for each example and do not learn a conditional distribution over feasible spatial hypotheses. In comparison, cSB ($K = 1$) reaches 87.3 even before introducing test-time multi-sample aggregation, outperforming both deterministic optimization baselines. This result provides evidence that distributional placement modeling itself offers an advantage beyond deterministic point estimation.

When $K = 5$, cSB further improves to 87.7 average accuracy, achieving 91.3, 75.8, 98.5, and 92.6 on General, Reasoning, OCR, and Doc&Chart, respectively. Relative to Deterministic Learned Placement, this corresponds to a 1.9-point improvement in average accuracy. Taken together, these results separate two complementary sources of improvement: the benefit of distributional placement modeling, already visible at $K = 1$, and an additional gain from retaining multiple spatial hypotheses at inference time.

(2) Distributional modeling and spatial constraints. We next disentangle the effects of stochasticity, spatial feasibility, and the specific cSB transport formulation. Stochastic Placement without a hard spatial constraint obtains an average accuracy of 84.3. Introducing projection onto the admissible region increases the average accuracy to 86.0, a gain of 1.7 points. The improvement is consistent across the four categories, with General, Reasoning, OCR, and Doc&Chart increasing

from 87.3, 70.9, 95.8, and 87.2 to 89.2, 73.4, 97.0, and 90.3, respectively. These results demonstrate the importance of explicitly preserving spatial feasibility for effective text-in-image prompting.

However, stochasticity combined with projection remains insufficient to match cSB. Stochastic + Projection reaches 86.0 average accuracy, whereas cSB achieves 87.7, yielding an additional 1.7-point improvement. At the category level, cSB improves General from 89.2 to 91.3, Reasoning from 73.4 to 75.8, OCR from 97.0 to 98.5, and Doc&Chart from 90.3 to 92.6. Thus, the gain cannot be attributed merely to injecting randomness and projecting the result back into a feasible region. Instead, learning a query-conditioned transport toward a structured target distribution provides an additional benefit.

We further compare cSB with a standard diffusion process equipped with the same type of spatial constraints. Standard Diffusion + Constraints obtains an average accuracy of 83.2, compared with 87.7 for cSB. Despite being both stochastic and constraint-aware, the standard diffusion baseline remains 4.5 points below cSB in average accuracy. This comparison indicates that generic stochastic diffusion dynamics alone are insufficient to explain the performance improvement; the distribution-matching structure introduced by cSB plays an important role.

Taken together, the ablations reveal a consistent pattern. Stochasticity alone does not improve upon the deterministic learned baseline, while explicit feasibility constraints substantially improve stochastic placement. cSB provides a further gain beyond constrained stochastic placement and standard constrained diffusion. This empirical trend is consistent with the distributional interpretation developed in Appendix A.

(3) Test-time spatial hypothesis sampling. A practical consequence of the distributional formulation is that multiple valid layouts can be sampled for the same image-query pair. We therefore study how the number of sampled spatial hypotheses K affects accuracy and computational cost.

With a single spatial hypothesis, cSB achieves an average accuracy of 87.3. Increasing the number of samples to $K = 3$ raises the score to 87.5, while $K = 5$ further improves it to 87.7. Larger values continue to provide modest gains, reaching 88.0 for $K = 7$ and 88.2 for $K = 10$.

The accuracy improvement exhibits diminishing returns as K increases. Relative to $K = 1$, the reported gains for $K = 3$, $K = 5$, $K = 7$, and $K = 10$ are 0.2, 0.4, 0.7, and 0.9 percentage points, respectively. In contrast, the computational cost grows approximately linearly with the number of samples. Latency increases from 1.84 seconds at $K = 1$ to 3.92 seconds at $K = 5$ and 7.85 seconds at $K = 10$, while computation increases from 4.6 GFLOPs to 10.5 and 21.0 GFLOPs, respectively. We therefore use $K = 5$ as the default setting, which provides a practical balance between accuracy and computational overhead.

Importantly, the gain from increasing K should not be interpreted simply as generic test-time augmentation. Each sample represents a distinct spatial hypothesis drawn from the learned conditional layout distribution. The consistent improvement as more hypotheses are retained therefore supports the value of modeling multiple feasible spatial configurations rather than collapsing the placement decision to a single deterministic layout.

Overall analysis. The controlled comparisons in Table 6 separate three sources of performance improvement. First, learned placement provides a flexible alternative to hand-designed or optimization-based placement strategies, although deterministic learning alone does not outperform all optimization baselines. Second, replacing a deterministic point predictor with cSB improves average accuracy from 85.8 to 87.3 even with a single sample ($K = 1$), demonstrating that the gain is not solely due to test-time multi-sample aggregation. Third, sampling multiple layouts from the learned distribution provides further improvements, reaching 87.7 at $K = 5$ and 88.2 at $K = 10$, albeit with increasing computational cost and diminishing accuracy gains.

Overall, these results support the view that cSB is more than a stochastic wrapper around a learned placement network. Its advantage arises from the joint effect of modeling a distribution over semantically compatible layouts, enforcing spatial feasibility, and retaining multiple plausible spatial hypotheses for test-time inference.

Table 6: Controlled analysis of deterministic and distributional spatial transport. We progressively isolate the effects of learned placement, distributional modeling, spatial constraints, and test-time stochastic sampling under matched settings.

(A) Matched Placement Comparison

Method	Learned	Stochastic	Constraint	General	Reasoning	OCR	Doc&Chart	Avg.
Semantic Greedy Placement	No	No	Yes	89.3	72.8	97.3	88.4	86.8
Direct Energy Minimization	No	No	Yes	89.7	72.1	96.2	88.5	86.5
Deterministic Learned Placement	Yes	No	Yes	88.5	71.8	96.4	88.0	85.8
cSB ($K = 1$)	Yes	Yes	Yes	90.1	74.1	97.8	91.0	87.3
cSB ($K = 5$)	Yes	Yes	Yes	**91.3**	**75.8**	**98.5**	**92.6**	**87.7**

(B) Distributional and Constraint Ablation

Method	Learned	Stochastic	Constraint	General	Reasoning	OCR	Doc&Chart	Avg.
Deterministic Learned Placement	Yes	No	Yes	88.5	71.8	96.4	88.0	85.8
Stochastic Placement	Yes	Yes	No	87.3	70.9	95.8	87.2	84.3
Stochastic + Projection	Yes	Yes	Yes	89.2	73.4	97.0	90.3	86.0
Standard Diffusion + Constraints	Yes	Yes	Yes	89.6	72.6	97.4	87.9	83.2
cSB (ours)	Yes	Yes	Yes	91.3	75.8	98.5	92.6	**87.7**

(C) Test-Time Spatial Sampling

Method	K	Avg. Acc.	Latency (s)	GFLOPs	Gain vs. $K = 1$	–	–	–
cSB	1	87.3	1.84	4.6	0.0	–	–	–
cSB	3	87.5	2.35	6.3	+0.2	–	–	–
cSB	5	**87.7**	3.92	10.5	**+0.4**	–	–	–
cSB	7	88.0	5.48	14.7	+0.7	–	–	–
cSB	10	88.2	7.85	21.0	+0.9	–	–	–

C Implementation Details

We train the cSB module for 100 epochs on 8 NVIDIA H20 GPUs with a batch size of 128 using AdamW, with an initial learning rate of 1×10^{-4}, weight decay of 1×10^{-2}, and a cosine annealing learning-rate schedule. The query-adaptive ink-budget coefficient is set to $\kappa = 80$. We use $\lambda_{\text{task}} = 1.0$ for the task-guided objective and $\lambda_{\text{sty}} = 1.0$ for the style regularization term, following the hyperparameter sensitivity results in Appendix D.3. The cSB sampler uses 20 refinement steps at inference. For test-time self-consistency, we independently sample K overlays for each image-query pair and obtain one prediction from the frozen MLLM for each resulting composite image; the final answer is determined by majority vote over the K predictions. We use $K = 5$ for the main experiments unless otherwise specified, and additionally report $K = 1$ results for single-sample and compute-matched comparisons. Timage is implemented in PyTorch 2.1 with Diffusers 0.24.0, using Qwen2.5-VL as the default frozen MLLM backbone.

D More Experimental Results

D.1 Mechanism Analysis of Timage

We investigate why Timage improves the performance of frozen vision–language models by disentangling the effects of semantic conditioning and spatial placement. As shown in Table 7, replacing the rendered query with a non-semantic string while preserving the same cSB-selected location causes the average accuracy to drop sharply from 87.7% to 57.1%, which is substantially below the base-image performance of 79.1%. This result indicates that the gain of Timage cannot be attributed to simply introducing an additional visual stimulus at a task-relevant location. Instead, inserting a non-semantic perturbation into the same region can substantially interfere with model reasoning.

We further compare Timage with a direct-overlay baseline, where the original query semantics are preserved but the query is rendered without the cSB-guided placement strategy. Direct overlay improves the average accuracy from 79.1% to 81.9%, showing that converting the textual query into a visual cue can itself provide useful conditioning. However, it remains 5.8 percentage points below Timage (81.9% vs. 87.7%), suggesting that semantic content alone does not fully explain the improvement. Together with the placement analysis presented earlier, these results suggest that Timage benefits from jointly preserving meaningful query semantics and placing them near task-relevant visual evidence.

Table 7: Mechanism intervention analysis of TIMAGE. We separately examine the effects of preserving query semantics and using cSB-guided spatial placement. A non-semantic replacement at the same cSB-selected location substantially degrades performance, while direct query overlay provides a moderate improvement over the base image. Combining meaningful query semantics with cSB-guided placement yields the best performance.

Intervention	Query Semantics	cSB-guided Placement	Avg. Acc. (%)
Base Image	✗	✗	79.1
Non-semantic Replacement	✗	✓	57.1
Direct Query Overlay	✓	✗	81.9
TIMAGE	✓	✓	**87.7**

Figure 7 provides a qualitative view of these interventions. For the spatial-relation query involving the TV and sofa, Timage produces a more concentrated response around the query-relevant visual regions. In contrast, replacing the query with a non-semantic string results in a more dispersed response and substantially lower accuracy. The direct-overlay baseline retains meaningful query semantics and therefore still exhibits responses around the relevant objects, but its response is less concentrated than that of Timage. These qualitative observations are consistent with the quantitative results in Table 7 and further support the hypothesis that the effectiveness of Timage arises from semantic–spatial coupling between the rendered query and task-relevant visual evidence, rather than from a generic local perturbation alone.



Figure 7: Qualitative mechanism analysis of semantic–spatial coupling in Timage. For a spatial-relation query involving the TV and sofa, Timage produces a concentrated response around the query-relevant visual regions. Replacing the rendered query with a non-semantic string at the same cSB-selected location leads to more dispersed responses and a substantial accuracy drop to 57.1%. Directly overlaying the meaningful query retains partial localization and achieves 81.9%, but remains below Timage at 87.7%.

D.2 ADDITIONAL GENERALIZATION EXPERIMENTS

We evaluate the generalization ability of Timage across model scales, benchmarks, and stronger MLLM backbones (Table 8). Timage consistently improves different MLLMs, ranging from lightweight models to large-scale backbones, demonstrating that its effectiveness does not rely on a particular model capacity. Moreover, the improvements across MMMU, MathVista, and DocVQA show that Timage generalizes beyond VMCBench to diverse multimodal reasoning and understanding scenarios. These results highlight that Timage learns a general input-space alignment mechanism rather than a benchmark-specific optimization strategy.

D.3 ROBUSTNESS TO INPUT AND RENDERING VARIATIONS

We further evaluate the robustness of Timage under several extreme variations in text rendering and input configurations. As shown in Table 9, the performance remains stable across all tested perturbations. Changing the font color from black to red results in only a minor accuracy change from 87.7% to 87.5%, suggesting that the effectiveness of Timage is not sensitive to a particular text color. Similarly, increasing the transparency of the rendered text leads to only a small decrease from 87.7% to 87.4%.

Table 8: Additional generalization experiments of Timage across ***model scales, benchmarks, and stronger MLLM backbones***.

Backbone	Size	Benchmark	Base	Timage	Δ
(a) Generalization across Model Scales					
Qwen2.5-VL-3B	3B	VMCBench	79.8	85.9	+6.1
Qwen2.5-VL-72B	72B	VMCBench	88.5	92.7	+4.2
(b) Generalization across Benchmarks					
Qwen2.5-VL-7B	7B	MMMU	58.6	61.6	+3.0
Qwen2.5-VL-7B	7B	MathVista	68.2	73.2	+5.0
Qwen2.5-VL-7B	7B	DocVQA	95.7	96.3	+0.6
(c) Generalization to Stronger and Larger MLLMs					
InternVL-2.5-26B	26B	VMCBench	90.1	93.5	+3.4
Qwen2.5-VL-32B	32B	VMCBench	91.2	94.4	+3.2
Qwen2-VL-72B	72B	VMCBench	86.5	91.9	+5.4

Table 9: Robustness of Timage to extreme variations in text rendering and input configurations. We evaluate changes in ***font color, transparency, query wording, and image resolution***. The consistently small performance variations indicate that Timage is relatively insensitive to these input perturbations.

Perturbation	Variant (Extreme Cases)	Acc. (%)
Font Color	Black (0,0,0) vs. Red (255,0,0)	87.7 vs. 87.5
Transparency	Opaque (1.0) vs. Highly Transparent (0.3)	87.7 vs. 87.4
Query Wording	Short Keyword vs. Long Description	87.7 vs. 87.3
Image Resolution	Low Res (72 DPI) vs. High Res (300 DPI)	87.2 vs. 87.9

We also examine variations in query wording and image resolution. Replacing short keyword-style queries with longer descriptions results in a modest change from 87.7% to 87.3%, indicating that the method remains effective under different levels of query verbosity. Moreover, varying the image resolution from 72 DPI to 300 DPI yields accuracies of 87.2% and 87.9%, respectively. Overall, these results demonstrate that Timage is robust to substantial variations in rendering appearance, query formulation, and input resolution, and does not rely on a narrowly tuned input.

D.4 Additional Hyperparameter Ablation Studies

We further investigate the sensitivity of Timage to several important hyperparameters, including the ink budget κ, the task optimization weight λ_{task}, the number of cSB refinement steps, and the style constraint weight λ_{style}. The results are summarized in Table 10.

First, we analyze the influence of the ink budget κ, which controls the amount of textual information rendered into the visual prompt. Timage achieves its best performance at $\kappa = 80$, reaching 87.7% accuracy. Both smaller and larger ink budgets lead to performance degradation. In particular, excessively large budgets introduce redundant visual patterns that may interfere with image understanding, while overly small budgets may provide insufficient textual information. These results suggest that a moderate amount of rendered text provides a favorable balance between information completeness and visual interference.

We next study the effect of the task optimization weight λ_{task}. Increasing λ_{task} from 0.1 to 1.0 improves the average accuracy from 86.4% to 87.7%, indicating that task-aware optimization provides useful guidance for generating effective visual prompts. However, further increasing λ_{task} to 2.0 slightly reduces the accuracy to 87.3%. This suggests that overly strong task supervision may introduce excessive optimization pressure and does not necessarily lead to better visual-language alignment. Overall, a moderate task loss weight provides the best trade-off.

Table 10: Hyperparameter sensitivity analysis of Timage.

Ink Budget κ	Setting	40	60	80	100	120
	Avg. Accuracy	86.8	85.6	87.7	85.5	83.1
λ_{task}	Setting	0.1	0.5	1.0	2.0	-
	Avg. Accuracy	86.4	87.2	87.7	87.3	-
cSB Steps	Setting	5	10	20	30	50
	Avg. Accuracy	84.2	85.9	87.7	86.5	84.1
λ_{style}	Setting	0	0.5	1.0	2.0	-
	Avg. Accuracy	83.8	84.9	87.7	87.4	-

We also examine the impact of the number of cSB refinement steps. Performance improves substantially as the number of steps increases from 5 to 20, with the accuracy rising from 84.2% to 87.7%. However, further increasing the number of steps leads to progressively lower performance, with accuracies of 86.5% and 84.1% at 30 and 50 steps, respectively. This indicates that sufficient refinement is important for generating effective visual prompts, whereas excessive refinement may introduce unnecessary deviations from favorable layouts.

Finally, we evaluate the contribution of the style constraint. Removing the style constraint ($\lambda_{style} = 0$) causes a substantial performance drop from 87.7% to 83.8%. Increasing λ_{style} to 0.5 improves the accuracy to 84.9%, while the best performance is achieved at $\lambda_{style} = 1.0$ with 87.7%. A further increase to 2.0 slightly reduces the accuracy to 87.4%. These results indicate that an appropriate style constraint is important for maintaining effective visual-language alignment, while excessively strong regularization can be counterproductive.

Overall, Timage exhibits a relatively stable performance across a broad range of hyperparameter settings, with moderate values generally providing the best performance. The results suggest that Timage does not rely on a narrowly tuned configuration.

D.5 Examples of Multilingual Inputs

We further examine the multilingual robustness of Timage on the **MME Landmark** benchmark. To evaluate ***whether the proposed text-in-image paradigm remains effective beyond English inputs***, we translate semantically equivalent queries into ten representative languages: English, French, Chinese, Korean, Japanese, Spanish, Arabic, Portuguese, Hindi, and German. These languages cover substantially different writing systems, including Latin scripts, Chinese characters, Hangul, Japanese Kana/Kanji, Arabic script, and Devanagari.

Figure 8 presents a representative example together with the average accuracy obtained for each language on MME Landmark. Timage achieves 95%, 93%, 94%, 90%, 88%, 90%, 88%, 88%, 85%, and 90% accuracy for English, French, Chinese, Korean, Japanese, Spanish, Arabic, Portuguese, Hindi, and German, respectively, corresponding to an average accuracy of 90.1% across the ten languages. Despite substantial differences in linguistic form and glyph appearance, Timage maintains consistently strong performance, suggesting that its effectiveness is not restricted to a particular language or alphabet.

Since a single Latin font such as Times New Roman does not provide complete glyph coverage for all writing systems, we use script-compatible fonts for multilingual rendering. Latin-script languages are rendered with the same Latin serif font, while Chinese, Japanese, Korean, Arabic, and Hindi use corresponding Unicode-compatible fonts with matched visual characteristics. For scripts requiring contextual shaping or bidirectional layout, such as Arabic and Devanagari, we use a Unicode-aware text shaping pipeline to ensure correct glyph composition and ordering. Importantly, the font file is the only language-dependent rendering component required for character coverage; the Timage placement strategy, ink-budget mechanism, transparency settings, optimization procedure, and all other rendering hyperparameters are kept unchanged across languages.

The example in Figure 8 further illustrates this behavior. The same landmark question is expressed in ten languages with markedly different visual forms, yet Timage consistently produces the correct prediction. These results indicate that Timage is robust not only to semantic variations across languages but also to substantial changes in the visual appearance of the rendered textual cue.



Figure 8: Multilingual robustness of Timage on the MME Landmark benchmark. We evaluate semantically equivalent queries in ten languages spanning multiple writing systems, including Latin, Chinese characters, Hangul, Japanese Kana/Kanji, Arabic, and Devanagari. Timage maintains consistently strong performance across languages, achieving an average accuracy of 90.1%. Script-compatible fonts are used only to ensure complete Unicode glyph coverage, while the remaining rendering and placement configurations are kept unchanged. The average accuracy for each language is shown at the bottom of the corresponding example.

E LIMITATIONS AND FUTURE WORK

As Figure 9 indicates, while Timage strikes a good performance-efficiency balance, two minor artifacts surface in extreme cases. First, the rendered cue may occasionally be partially incomplete (e.g., a missing stroke); our experiments show this rarely dents reasoning accuracy. Second, in heavily textured or low-contrast areas the overlay's transparency can be suboptimal, letting the cue blend into the background. Future work will refine the renderer to ensure both the completeness and the distinctiveness of the cue without sacrificing efficiency.



Figure 9: Failure cases of our method.