GENERATIVE MULTI-MODAL GRADIENT MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-modal Large Language Models (MLLMs) rely on massive web-scraped corpora that are noisy, redundant, and weakly aligned with downstream objectives, leading to persistent failure modes such as object hallucination and brittle compositional generalization. Conventional text-to-image (T2I) synthesis offers a scalable alternative but remains *passive*: prompts drive sampling without feedback on training utility, producing images that are visually plausible yet redundant for fine-tuning. We present **Generative Multi-modal Gradient Matching (GMGM)**, an active synthesis framework that, given a frozen T2I generator and a small real reference batch, optimizes per-prompt latent codes so that each generated image is simultaneously: (i) prompt-faithful, (ii) feature-aligned with real data via Sparse Autoencoder (SAE) distribution matching, and (iii) gradient-informative for downstream MLLM fine-tuning. Our key insight is that synthetic data should be optimized not merely for perceptual realism, but for alignment with the downstream model's learning dynamics. We prove that minimizing this tri-objective tightens the upper bound on the generalization gap, isolating covariate shift, feature coverage, and trajectory divergence. Empirically, relative to a strong prompt-matched synthetic baseline (TreeSynth), GMGM improves hallucination robustness (+5.6% POPE F1) and compositional reasoning (+58.1% Winoground), while matching its full-budget performance using only **10%** of the synthetic data budget. Further experiments demonstrate GMGM's strong cross-architecture generalization.

1 INTRODUCTION

The rapid progress of Multi-modal Large Language Models (MLLMs) Liu et al. (2023); Alayrac et al. (2022) has greatly advanced visual–language reasoning. However, their performance remains fundamentally constrained by the quality of their training data. Large-scale web corpora such as LAION-5B Schuhmann et al. (2022) offer broad coverage, but are inherently noisy, biased, and skewed toward coarse-grained patterns, while underrepresenting compositional and fine-grained visual concepts. This mismatch between pretraining data and downstream objectives contributes directly to persistent failure modes, including object hallucination Li et al. (2023b) and brittle compositional generalization Thrush et al. (2022). As simply scaling raw data yields diminishing returns, a central challenge is no longer how to collect more data, but how to construct data that is more useful for downstream adaptation. In this work, we ask: *given a frozen text-to-image (T2I) generator $\mathcal{G}$ and a small real reference batch (e.g., 50 images per category), how can we synthesize a training set $\mathcal{D}_{syn}$ that maximizes downstream MLLM fine-tuning performance?*

A natural solution is to use modern T2I models such as Stable Diffusion Rombach et al. (2022) to generate training images at scale. Yet existing T2I synthesis pipelines are inherently *passive*: prompts determine what is sampled, but the generation process receives no feedback from the downstream learner. As a result, synthesized images may be semantically plausible while remaining uninformative for training, often collapsing into high-density regions of the generator that induce weak, redundant, or misaligned gradients during fine-tuning. This reveals a key **utility gap** between visual realism and training effectiveness: for synthetic data to improve MLLMs, it must do more than look realistic or match prompts. It must also cover underrepresented features and produce learning signals that meaningfully shape the model's parameters.

Existing approaches address this challenge only in part. Prompting-based methods such as TreeSynth Wang et al. (2025) improve semantic diversity, but treat the generator largely as a black

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.

box and provide little control over feature-level coverage or downstream optimization behavior. On the other hand, dataset condensation and trajectory-matching methods, such as SRe^{2}L Yin et al. (2023) and LGM Cazenavette et al. (2022), explicitly optimize synthetic data for training utility, but were developed for unconditional vision settings and lack the semantic controllability required for MLLM fine-tuning. More broadly, existing methods tend to optimize only one aspect of useful synthesis, such as semantic diversity, feature alignment, or gradient utility, without unifying these aspects within a single framework.

To close this gap, we propose **Generative Multi-modal Gradient Matching (GMGM)**, an instance-level latent optimization framework that converts the generator into an active search engine for high-utility supervision. Guided by a theoretical generalization bound, GMGM enforces three complementary properties simultaneously: (1) **semantic consistency** with the prompt; (2) coverage of **disentangled visual features** through a frozen Sparse Autoencoder (SAE) Bricken et al. (2023); Cunningham et al. (2023), which decomposes dense representations into sparse and individually interpretable feature directions; and (3) **gradient informativeness** measured against real data through a randomized linear proxy probe. By coupling a frozen MLLM critic, an SAE-based feature controller, and the proxy probe, GMGM transforms passive sampling into a feedback-driven optimization process. Our contributions are as follows:

- **Unified active synthesis framework.** GMGM jointly enforces semantic alignment, feature coverage in SAE space, and gradient informativeness within a single latent optimization objective, thereby addressing the key limitations of passive T2I synthesis.
- **Theoretical justification.** We derive a three-term generalization bound that links each objective to a distinct source of risk, namely covariate shift, feature coverage gap, and one-step trajectory divergence, showing that GMGM provably reduces the synthetic-to-real gap and providing principled support for its design.
- **Strong empirical gains and data efficiency.** Across six benchmarks and three VLM backbones, GMGM already matches the full-budget performance of our strong synthetic baseline (TreeSynth) using only **10%** of the synthetic data budget, and delivers further gains at full scale, including +58.1% on Winoground and +5.6% POPE F1 over that baseline.

2 Related Work

2.1 MLLM Fine-tuning and Data Quality

The evolution of MLLMs has progressed from contrastive vision–language pretraining Radford et al. (2021) to advanced systems that bridge vision encoders with frozen LLMs, including Flamingo Alayrac et al. (2022), BLIP-2 Li et al. (2023a), and LLaVA Liu et al. (2023). While recent efforts emphasize architectural scaling and instruction diversity Zhang et al. (2026), downstream performance remains tightly coupled to the quality of fine-tuning data. Web-scraped corpora such as LAION-5B Schuhmann et al. (2022) provide broad coverage but are noisy and lack fine-grained compositional structure, directly contributing to object hallucination Li et al. (2023b) and weak compositionality Thrush et al. (2022). Recent practice has therefore turned to data *curation*: filtering pipelines (DataComp Gadre et al. (2023)), recaptioning (Qwen3-VL Bai et al. (2025)), and synthetic supervision (MAmmoTH-VL Guo et al. (2025)). These methods improve language-side or alignment quality, but operate on existing image distributions and do not address feature-level coverage or gradient informativeness on the visual side, which GMGM targets directly.

2.2 Synthetic Data Generation for MLLMs

T2I models such as Stable Diffusion Rombach et al. (2022) and DALL-E Ramesh et al. (2022) enable scalable synthesis of multimodal data. However, most pipelines remain *passive*: even structural prompting like TreeSynth Wang et al. (2025) treats the generator as a black box, optimizing for prompt-level diversity without visibility into downstream training utility. Generated samples consequently concentrate in high-density modes of the generator and provide redundant gradient signal during fine-tuning. While feedback-driven mechanisms have begun to emerge Sun et al. (2025), no prior framework jointly enforces semantic alignment, structured feature coverage, and training-

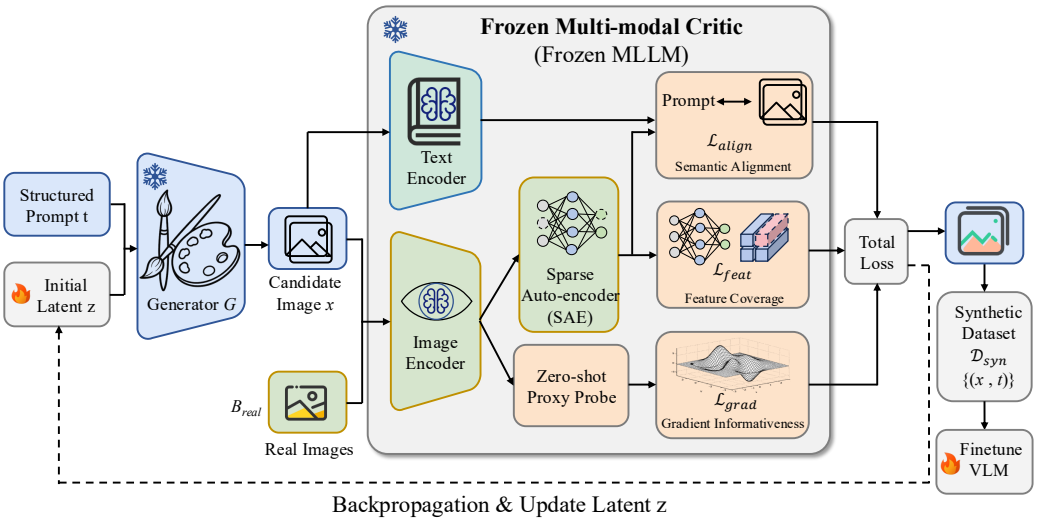

Figure 1: **The GMGM Framework.** A frozen multi-modal critic guides the optimization of synthetic images $x = \mathcal{G}(z; t)$ by balancing semantic alignment ($\mathcal{L}_{align}$), SAE-space feature distribution matching ($\mathcal{L}_{feat}$), and proxy-gradient alignment to a real reference batch $\mathcal{B}_{real}$ ($\mathcal{L}_{grad}$). The optimized latent codes yield a high-utility synthetic dataset $\mathcal{D}_{syn}$ for efficient MLLM fine-tuning.

aware optimization. GMGM fills this gap with a critic-guided latent optimization loop, thereby turning synthetic data generation from passive sampling into downstream-aware supervision design.

2.3 Dataset Distillation and Gradient Matching

Dataset distillation Wang et al. (2018) compresses large datasets into compact synthetic sets that preserve training dynamics. Bilevel approaches have evolved into more stable gradient Zhao et al. (2020) and trajectory matching Cazenavette et al. (2022) techniques, recently extended to large pre-trained models via linear probes. Direct extension to MLLMs is challenging: gradient alignment alone is insufficient when the synthetic samples must also respect fine-grained textual intent. GMGM adapts this principle to the multimodal setting by (i) operating in the structured, monosemantic feature space of a frozen Sparse Autoencoder Pach et al. (2025), and (ii) using a randomized linear proxy on the shared frozen vision encoder to obtain a stable, non-vanishing gradient utility signal aligned with downstream visual representations.

3 Methodology

In this section, we introduce **Generative Multi-modal Gradient Matching (GMGM)**, an instance-level latent optimization framework that converts a frozen T2I generator into an active source of high-utility supervision. Notation is summarized in Table A1 and the full procedure in Algorithm 1.

3.1 Problem Setup

Inputs. GMGM assumes access to: (i) a frozen T2I generator $\mathcal{G}$ that maps a latent code z and a text prompt t to an image $x = \mathcal{G}(z; t)$; (ii) a small real reference set $\{\mathcal{B}^{(c)}_{real}\}_c$ containing a few dozen images per super-category y_c, used *only* as a feature/gradient anchor (not for downstream fine-tuning); (iii) a frozen multi-modal critic comprising a text encoder $\mathcal{E}_T$, a vision encoder $\mathcal{E}_V$, a Sparse Autoencoder (SAE) $S(\cdot)$ on top of $\mathcal{E}_V$, and a randomly initialized linear proxy probe ϕ.

Output. A synthetic dataset $\mathcal{D}_{syn} = \{(x^*_i, t_i)\}^N_{i=1}$ used to fine-tune a downstream MLLM, where each $x^*_i = \mathcal{G}(z^*_i; t_i)$ is obtained by per-prompt latent optimization.

Objective. For each prompt t, we want $x = \mathcal{G}(z;t)$ to simultaneously (i) preserve prompt semantics, (ii) match the interpretable feature distribution of $\mathcal{B}_{real}$, and (iii) induce a training update comparable to that of real samples. We solve the following instance-level objective, where $x = \mathcal{G}(z;t)$ is implicit in all losses:

$$z^* = \arg\min_z \ \lambda_{align}\mathcal{L}_{align}(z;t) \ + \ \lambda_{feat}\mathcal{L}_{feat}(z;\mathcal{B}_{real}) \ + \ \lambda_{grad}\mathcal{L}_{grad}(z;\mathcal{B}_{real}). \quad (1)$$

The optimized image $x^* = \mathcal{G}(z^*;t)$ is then added to $\mathcal{D}_{syn}$. Repeating this procedure over prompts constructs a high-utility synthetic dataset.

3.2 GMGM: Critic-Guided Latent Optimization

The objective in Eq. 1 decomposes into three losses, each targeting one failure mode of passive T2I synthesis: *semantic drift* (handled by $\mathcal{L}_{align}$), *feature mode collapse* (handled by $\mathcal{L}_{feat}$), and *redundant gradients* (handled by $\mathcal{L}_{grad}$). We describe each in turn.

Semantic alignment loss ($\mathcal{L}_{align}$). We first enforce prompt-faithful generation using frozen text and vision encoders $\mathcal{E}_T$ and $\mathcal{E}_V$. Let $u = \mathcal{E}_T(t)$ and $v = \mathcal{E}_V(x)$ where $x = \mathcal{G}(z;t)$. We use cosine distance:

$$\mathcal{L}_{align}(z;t) = 1 - \frac{\langle \mathcal{E}_T(t), \mathcal{E}_V(\mathcal{G}(z;t)) \rangle}{\|\mathcal{E}_T(t)\|_2 \cdot \|\mathcal{E}_V(\mathcal{G}(z;t))\|_2}. \quad (2)$$

SAE feature distribution matching ($\mathcal{L}_{feat}$). Semantic alignment alone admits many visually plausible solutions that nevertheless concentrate in a narrow region of the visual feature space, producing redundant supervision. We therefore additionally constrain the *visual feature distribution* of synthesized images to track that of real data. We do so in a Sparse Autoencoder (SAE) space: a frozen Top-K SAE Pach et al. (2025) pre-trained on natural images decomposes the dense vision-encoder activations into a high-dimensional, sparse, and approximately monosemantic feature basis Bricken et al. (2023); Cunningham et al. (2023). Each active SAE coordinate corresponds to a single interpretable visual concept (e.g., a texture, object part, or relation), so distribution matching in this space directly enforces *concept-level* coverage, not pixel similarity. Directly matching a single real-image feature would cause copying and mode collapse; instead, GMGM matches the *batch-level* real feature distribution. Let $s(x) = S(x) \in \mathbb{R}^m$ denote the sparse activation vector. We use a Maximum Mean Discrepancy (MMD) objective, a kernel-based two-sample distance:

$$\mathcal{L}_{feat}(z;\mathcal{B}_{real}) = \mathrm{MMD}^2(\{s(\mathcal{G}(z;t))\}, \{s(x_r)\}_{x_r \in \mathcal{B}_{real}}), \quad (3)$$

where $\mathrm{MMD}^2(\cdot,\cdot)$ is computed with a Gaussian kernel in SAE feature space.[1]

Gradient matching via randomized proxy probe ($\mathcal{L}_{grad}$). Distribution matching ensures *what* features appear, but not whether they will be informative for downstream parameter updates. We therefore add a gradient-level objective that prefers samples whose induced parameter update resembles that of real data. A naive implementation—matching gradients on the full MLLM parameters—is both prohibitively expensive and prone to *gradient collapse*: when the reference network is already confident, $\bar{g}_{real} \to 0$ and the matching loss becomes vacuous. GMGM avoids both issues with a *randomized linear proxy probe* ϕ on top of the frozen vision encoder $\mathcal{E}_V$. At every latent-optimization run, we sample fresh probe parameters $\phi \sim \mathcal{N}(0, \sigma^2 I)$, which keeps the probe in a high-entropy state with non-vanishing gradients. The proxy loss is the cross-entropy $\ell(\phi(\mathcal{E}_V(x)), y_c)$ against the super-category label y_c (used only here for gradient computation, not for downstream supervision).

We compute the synthetic gradient with respect to proxy parameters:

$$g_{syn}(z) = \nabla_\phi \, \ell(\phi(\mathcal{E}_V(\mathcal{G}(z;t))), y_c), \quad (4)$$

and the mean real gradient:

$$\bar{g}_{real} = \frac{1}{|\mathcal{B}_{real}|} \sum_{x_r \in \mathcal{B}_{real}} \nabla_\phi \, \ell(\phi(\mathcal{E}_V(x_r)), y_c). \quad (5)$$

[1] Any characteristic kernel is valid; we use a Gaussian kernel in practice.

Algorithm 1: GMGM: Critic-guided latent optimization for one prompt

Input: prompt t, real batch $\mathcal{B}_{\text{real}}$ with label y_c, generator $\mathcal{G}$, encoders $\mathcal{E}_V, \mathcal{E}_T$, SAE feature $S(\cdot)$
Hyperparams: weights $\lambda_{align}, \lambda_{feat}, \lambda_{grad}$, steps K, step size α
Sample $z_0 \sim \mathcal{N}(0, I)$; sample randomized probe $\phi \sim \mathcal{N}(0, \sigma^2 I)$
Compute $\bar{g}_{\text{real}}$ using (5)
for $k = 0$ to $K - 1$ **do**
    $x_k = \mathcal{G}(z_k; t)$
    Compute $\mathcal{L}_{\text{align}}$ by (2), $\mathcal{L}_{\text{feat}}$ by (3), $\mathcal{L}_{\text{grad}}$ by (6)
    Update z_{k+1} using (7)
end for
Output: $x^* = \mathcal{G}(z_K; t)$

We then align these update directions using cosine distance:

$$\mathcal{L}_{grad}(z; \mathcal{B}_{real}) = 1 - \frac{\langle g_{syn}(z), \bar{g}_{real} \rangle}{\|g_{syn}(z)\|_2 \cdot \|\bar{g}_{real}\|_2 + \epsilon}. \tag{6}$$

This objective prefers synthetic instances whose induced parameter update direction resembles that of real data, prioritizing samples with high training utility. Crucially, since ϕ is a small linear head sharing the frozen $\mathcal{E}_V$ used by downstream MLLMs, minimizing $\mathcal{L}_{grad}$ aligns the proxy update direction with that induced by real data, which Appendix B formalizes as first-order trajectory alignment; we further verify empirically that this proxy ranks samples consistently with the downstream LoRA update direction (Appendix C).

Latent update. Combining the three terms, GMGM starts from $z_0 \sim \mathcal{N}(0, I)$ and iteratively updates the latent code for K steps:

$$\begin{aligned} z_{k+1} = z_k - \alpha \nabla_z \Big[& \lambda_{align} \mathcal{L}_{align}(z_k; t) + \lambda_{feat} \mathcal{L}_{feat}(z_k; \mathcal{B}_{real}) \\ & + \lambda_{grad} \mathcal{L}_{grad}(z_k; \mathcal{B}_{real}) \Big]. \end{aligned} \tag{7}$$

The final image $x^* = \mathcal{G}(z_K; t)$ is stored as a synthetic training example.

3.3 Theoretical Analysis of the Generalization Gap

We show that the GMGM objective (Eq. 1) tightens an upper bound on the generalization gap between models trained on $\mathcal{D}_{\text{syn}}$ and evaluated on P_{real}. Under mild regularity conditions—a Lipschitz frozen generator, bounded Lipschitz encoders, an L_S-Lipschitz sparse SAE ($\|s(x)\|_0 \le K_0 \ll m$), bounded proxy gradients, and a characteristic MMD kernel (Assumptions 1–6)—the gap decomposes into three interpretable terms, each controlled by one GMGM loss.

Theorem 1 (Three-Term Generalization Bound)**.** *Under Assumptions 1–6, for any hypothesis $f_\theta \in \mathcal{F}$ trained on synthetic data $\mathcal{D}_{syn} \sim P_{syn}$, the population risk on P_{real} satisfies:*

$$\begin{aligned} R(f_\theta; P_{\text{real}}) \;\le\; & R(f_\theta; P_{\text{syn}}) && \text{(training risk)} \\ & + \underbrace{C_1 \cdot \mathcal{L}_{\text{align}}}_{\text{(I): semantic shift}} + \underbrace{C_2 \cdot \Delta_S(P_{\text{syn}}, P_{\text{real}})^2}_{\text{(II): feature coverage gap}} \\ & + \underbrace{C_3 \cdot \mathbb{E}_z\big[\mathcal{L}_{\text{grad}}(z; \mathcal{B}_{\text{real}})\big]}_{\text{(III): trajectory divergence}} + \lambda^*_{\text{opt}}, \end{aligned}$$

where $C_1 = 2MB_hB_u$, $C_2 = ML_S^2$, $C_3 = 2MB_g$, and $\lambda^*_{opt} = \min_{f \in \mathcal{F}}[R(f; P_{real}) + R(f; P_{syn})]$ is the optimal joint error.

Each loss governs one source of synthetic-to-real risk: $\mathcal{L}_{\text{align}}$ bounds the covariate shift (Term I) through the $\mathcal{H}\Delta\mathcal{H}$-divergence in encoder space Ben-David et al. (2010); the MMD loss $\mathcal{L}_{\text{feat}}$ controls the feature-coverage gap (Term II) by matching all moments in the sparse SAE space; and $\mathcal{L}_{\text{grad}}$ aligns the one-step update direction (Term III), where re-sampling the probe $\phi \sim \mathcal{N}(0, \sigma^2 I)$ keeps the expected squared gradient norm strictly positive and thus prevents gradient collapse. The latent optimization converges to a stationary point at rate $\mathcal{O}(K^{-1/2})$. Full assumptions, lemmas, and proofs are deferred to Appendix B.

Table 1: Comprehensive performance comparison of LLaVA-1.6-7B fine-tuned on 50k samples (100%). *NAvg.* is the zero-shot-normalized average ($\text{score}/\text{score}_{\text{zs}} \times 100$). Δ denotes the relative improvement of GMGM^{SDXL} (100%) over TreeSynth. *IN21k-Cap* (R1) and *LLaVA-150K* (R2) are instruction-aligned real-data baselines (Appendix E.1); *FLUX.2-Base/+GMGM* swap the underlying T2I generator (Appendix E.2). Top-3 are **red**, **blue**, **green**.

	Method	Winoground	POPE	VQAv2	GQA	MME-Cog.	MMBench	NAvg.
Real	Zero-shot	9.8	82.5	76.5	45.2	298.4	59.8	100.00
	Real Random	12.1	83.6	78.4	47.1	305.2	61.8	106.19
	Real Coreset	13.2	84.8	79.5	48.6	312.8	63.5	109.99
	IN21k-Cap (R1)	14.5	85.3	80.1	49.2	316.4	64.2	112.51
	LLaVA-150K Subset (R2)	15.1	85.8	80.6	49.7	319.1	64.9	114.23
Generative	SDXL-Base	10.9	81.9	77.6	46.1	301.2	60.5	102.67
	InstructPix2Pix	12.1	83.5	78.8	47.3	310.5	62.1	106.71
	TreeSynth	12.4	83.8	79.2	47.8	315.4	62.6	107.96
	SRe2L	13.8	84.5	79.8	48.5	318.5	63.1	111.18
	ShareGPT4V	14.2	85.1	80.5	49.6	321.8	64.5	113.12
Opt.	LGM	15.8	85.2	80.2	49.1	318.2	63.8	115.21
	Feat-Steering	16.2	87.5	80.8	49.9	325.6	64.2	117.31
Stronger generator (FLUX.2)								
	FLUX.2-Base	14.8	85.2	80.1	48.3	318.5	63.8	112.84
	FLUX.2 + GMGM (100%)	**23.4**	**90.2**	**85.4**	**56.7**	**368.2**	**71.4**	**136.18**
	GMGM (1%)	10.5	83.5	78.1	46.5	304.5	61.2	102.95
	GMGM (5%)	10.9	84.4	79.0	47.4	311.2	62.1	104.97
	GMGM (10%)	12.8	85.1	79.9	48.8	320.1	62.8	109.74
	GMGM (50%)	**16.8**	**87.9**	**82.5**	**52.4**	**340.5**	**67.4**	**121.43**
	GMGM (100%)	**19.6**	**88.5**	**84.1**	**54.8**	**358.9**	**69.1**	**129.05**
	Δ vs TreeSynth	+58.1%	+5.6%	+6.2%	+14.6%	+13.8%	+10.4%	+19.5%

4 Experiments

We design experiments to answer four questions: **(Q1)** Does GMGM outperform synthetic and real-data baselines (Section 4.2)? **(Q2)** How does it scale with data and compute (Section 4.3)? **(Q3)** Which loss components drive the gains (Section 4.4)? **(Q4)** Do the gains transfer across VLM backbones (Section 4.6)?

4.1 Experimental Setting

Benchmarks. We evaluate on six benchmarks spanning complementary multimodal skills: (i) *general VQA*: VQAv2 Antol et al. (2015) and GQA Hudson & Manning (2019) for grounding and relational reasoning, and Winoground Thrush et al. (2022) for fine-grained vision–language compositionality; (ii) *hallucination robustness*: POPE for object-existence verification; (iii) *holistic perception and cognition*: MME Fu et al. (2023) (Cognition subset) and MMBench Liu et al. (2024) (bilingual multi-choice) for higher-level reasoning under distractors. Details are in Appendix D.

Comparison methods. We compare GMGM against three families of baselines. *Real-data baselines: Real Random* (uniform 10% sample of ImageNet-21k), *Real Coreset* Sener & Savarese (2017), plus two instruction-aligned references that match GMGM's modality: *IN21k-Cap (R1)* (Qwen2-VL-72B captions over the same 50/category stratum) and *LLaVA-150K Subset (R2)* (CLIP-stratified 50/category slice of LLaVA-Instruct-150K); see Appendix E.1. *Generative baselines:* SDXL-Base Podell et al. (2023), InstructPix2Pix Brooks et al. (2023), TreeSynth Wang et al. (2025), SRe^{2}L Yin et al. (2023), ShareGPT4V Chen et al. (2024), and *FLUX.2-Base* Labs (2025) (a stronger 2025-era T2I generator, integrated under a 2×2 generator$\times$method protocol; Appendix E.2). *Optimization-baseline ablations:* LGM (gradient matching without SAE feature control) and Feat-Steering (SAE feature steering without gradient matching).

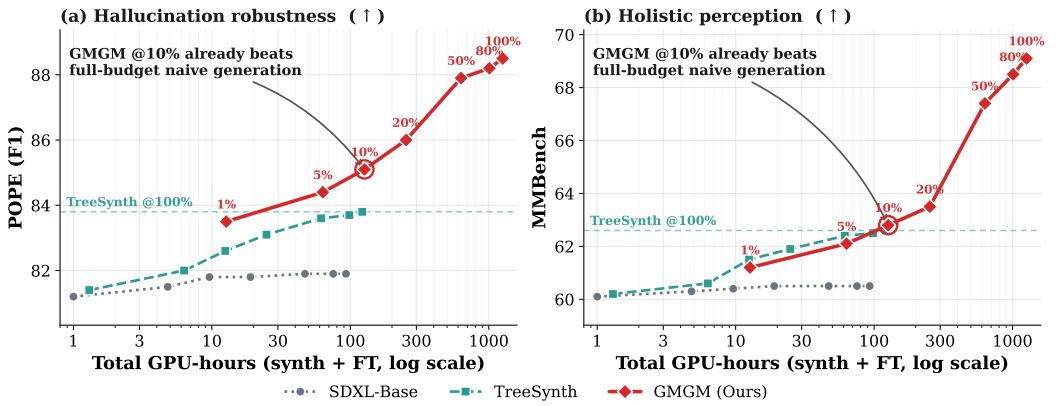

Figure 2: **Compute-matched Pareto comparison.** x-axis: total GPU-hours (synthesis + fine-tuning, log scale); markers denote $[1, 5, 10, 20, 50, 80, 100]\%$ of the full dataset, shown for **(a)** POPE and **(b)** MMBench. SDXL-Base and TreeSynth saturate early, whereas GMGM keeps improving and already surpasses the full-budget naive-generation setting at only a 10% budget.

Implementation details. The default frozen generator is SDXL-1.0 Podell et al. (2023); under the 2×2 generator$\times$method protocol (Appendix E.2) we additionally use FLUX.2 Labs (2025), with the latent re-projected to its native shape and the Lipschitz constant L_G (Asm. 1) re-estimated on a 1k-prompt slice. The frozen multi-modal critic uses the CLIP ViT-L/14 tower of LLaVA-1.6-7B; InternVL2-8B, Qwen2-VL-7B, and Qwen3-VL-7B-Instruct Bai et al. (2025) are evaluated as transfer targets. Per-prompt latent optimization runs $K{=}50$ AdamW steps with forward-mode JVP; loss weights are $\lambda_{align}{=}10$, $\lambda_{feat}{=}1$, $\lambda_{grad}{=}50$, and the SAE is a frozen Top-K autoencoder ($K{=}32$). Downstream fine-tuning applies LoRA to all linear projections (one epoch of AdamW on $8\times$H100); the Qwen3-VL-7B matrix uses three seeds per cell with paired-bootstrap CIs. Full recipes are in Appendices E.2–E.3.

4.2 Main Results

We first answer **Q1**. Table 1 reports LLaVA-1.6-7B fine-tuned on each data source across all six benchmarks: **GMGM dominates every metric** under both generators. The largest gains fall on our objective's intended failure modes, compositional reasoning and hallucination robustness: GMGM^{SDXL} improves over the strong text-steered baseline TreeSynth by **+58.1%** on Winoground and attains the best POPE among all methods, surpassing even real-data coreset selection, while holistic perception (MME-Cog., MMBench) improves in step.

Robustness to baseline curation and generator strength. We address two concerns (that raw ImageNet-21k handicaps the real-data baselines on *language modality*, and that GMGM merely compensates for SDXL-1.0's compositional weaknesses) with an instruction-aligned real-data control (R1, R2) and a fully-crossed 2×2 generator$\times$method design (SDXL-1.0 vs. FLUX.2). Both preserve GMGM's ordering: the instruction-aligned references raise the real-data baseline yet remain well below GMGM^{SDXL}, and GMGM's headroom shrinks but persists under the stronger generator. This isolates the residual gain as *gradient informativeness* rather than instruction format or generator quality (Appendices E.1–E.2).

The data-efficiency story is equally striking: with only **10%** of the data budget GMGM already surpasses TreeSynth trained at full budget, and by **50%** it overtakes the strongest non-GMGM competitor (Table 1). Gradient-aware synthesis thus converts each sample into substantially denser supervision than passive generation or coreset selection.

4.3 Data and Compute Efficiency

We answer **Q2** along two axes: *data efficiency* and *compute efficiency*. GMGM exhibits a *steeper* data-scaling curve than real-data and naive generative baselines on all six benchmarks (per-

Table 2: Component Ablation and Hyperparameter Sensitivity.

Config	$\mathcal{L}_{align}$	$\mathcal{L}_{feat}$ (λ)	$\mathcal{L}_{grad}$ (λ)	Winoground	POPE (F1)	MMBench
A	✓	-	-	11.5	82.1	61.0
B	✓	**1.0**	-	14.1	88.1	63.5
C	✓	**0.1**	-	12.2	84.5	61.8
D	✓	**10.0**	-	13.5	87.2	62.9
E	✓	-	**50**	13.2	84.5	64.2
F	✓	-	**10**	12.0	83.2	62.1
G	✓	-	**100**	12.8	83.8	63.5
H	✓	**1.0**	**50**	**19.6**	**88.5**	**69.1**

benchmark curves in Fig. A1), with the largest gaps on reasoning-heavy settings where standard T2I sampling and coreset selection saturate early.

Compute-matched efficiency. To check that the gain is not simply more compute, Fig. 2 sweeps budgets from 1%–100% and plots performance against *total GPU-hours* (latent optimization + generation + fine-tuning), so every point is compute-matched. Naive T2I pipelines (SDXL-Base, TreeSynth) plateau quickly once the generator's high-density modes are covered. GMGM keeps improving across the entire range and dominates the Pareto frontier at every budget; under only a 10% budget it already matches the total compute of and surpasses the full-budget naive-generation setting, indicating substantially higher information density per sample. A per-image step-count sweep (Appendix G) confirms the budget is well chosen: downstream utility is near-saturated by the default K=50, with K=30 offering a $\sim$1.7$\times$ cheaper operating point at >99% of the utility.

4.4 Ablation Study

To answer **Q3**, we sweep each loss weight in Table 2. Feature matching peaks at λ_{feat}=1.0 (Config B) and gradient matching at λ_{grad}=50 (Config E); the full configuration (Config H) is best on every benchmark, confirming the two signals are *complementary*: $\mathcal{L}_{feat}$ enriches *what* is synthesized while $\mathcal{L}_{grad}$ prioritizes *which* samples help the downstream update.

4.5 Does the Proxy Gradient Predict Downstream Utility?

Because the randomized linear proxy is optimized in a parameter space disjoint from the downstream LoRA adapter, we assess directly whether its gradient signal is predictive of downstream fine-tuning utility (full protocol and tables in Appendix C). At the *instance* level, the GMGM proxy-alignment score exhibits a Spearman rank correlation of 0.65 (95% CI $[0.62, 0.68]$) with the true LoRA-update alignment on held-out candidates, substantially exceeding both CLIP semantic alignment (0.18) and the SAE feature score (0.35), with mutually non-overlapping confidence intervals. At the *dataset* level, stratifying candidates by proxy alignment and fine-tuning on compute-matched top-, random-, and bottom-quartile subsets induces the hypothesized monotonic ordering in aggregate performance (NAvg. $112.9 > 109.7 > 103.2$). These results establish that $\mathcal{L}_{grad}$ preferentially selects samples of genuinely higher downstream utility, rather than exploiting idiosyncrasies of the frozen critic.

4.6 Cross-Architecture Transfer

To answer **Q4**, we transfer the same GMGM-synthesized dataset to LLaVA-1.6-7B, InternVL2-8B, Qwen2-VL-7B, and the 2025-era Qwen3-VL-7B-Instruct. Table 3 shows consistent gains across all four architectures over both zero-shot and naive SDXL-based synthesis on Winoground, POPE, and MMBench. The bottom block additionally reports a fully-crossed five-cell matrix (C0–C4) on Qwen3-VL-7B over generators$\times$methods, averaged over three seeds (Appendix E.3). The pre-registered monotonic ordering C0 $<$ C1 $\leq$ C2 $<$ C3 $\leq$ C4 holds on every benchmark, with C4 (FLUX.2 + GMGM) best throughout, and a paired bootstrap test confirms all four headline contrasts (Bonferroni-corrected, all $p < 0.0125$; Tab. A1). This separates base perceptual quality from gradient informativeness, supporting Theorem 1 and indicating that GMGM provides broadly useful

Table 3: Cross-architectural transfer of GMGM training data. The bottom block reports the Qwen3-VL-7B five-cell matrix (Appendix E.3), fully crossing generator (SDXL-1.0 vs. FLUX.2) with method (passive vs. GMGM). Each Qwen3-VL-7B condition is averaged over 3 seeds; paired bootstrap CIs are reported in Tab. A1.

MLLM models	Training Data	Winoground	POPE (F1)	MMBench
LLaVA-1.6-7B	Zero-shot	9.8	82.5	59.8
	SDXL-Base (100%)	10.9	81.9	60.5
	GMGM (100%)	**19.6**	**88.5**	**69.1**
InternVL2-8B	Zero-shot	13.5	84.6	65.5
	SDXL-Base (100%)	14.1	84.2	66.1
	GMGM (100%)	**19.8**	**90.1**	**70.4**
Qwen2-VL-7B	Zero-shot	12.2	83.8	63.8
	SDXL-Base (100%)	13.0	83.5	64.2
	GMGM (100%)	**18.5**	**89.2**	**68.5**
Qwen3-VL-7B	C0: Zero-shot	14.2	84.9	66.2
	C1: + SDXL-Base (passive)	15.4	85.4	67.1
	C2: + FLUX.2-Base (passive)	18.2	87.1	69.4
	C3: + SDXL + GMGM	21.5	89.6	71.8
	C4: + FLUX.2 + GMGM	**26.8**	**91.5**	**74.5**

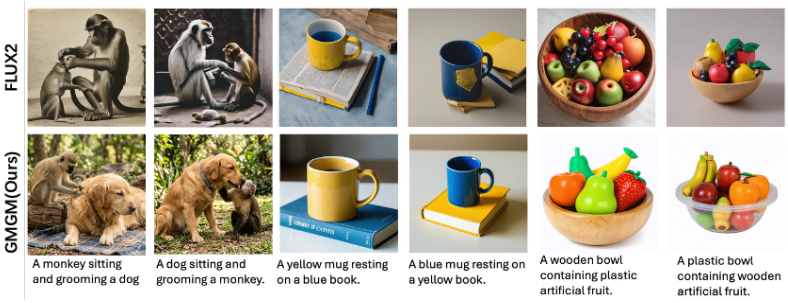

Figure 3: Visualization of generated samples from our GMGM and the SDXL/FLUX.2 baselines.

supervision. GMGM also transfers across *critics*: a matched SigLIP critic preserves the gains and a plug-in OCR module corrects a known blind spot (Appendix H).

4.7 Qualitative Analysis

Figure 3 compares SDXL and GMGM samples. SDXL exhibits *relation reversal* (e.g., "a dog grooming a monkey") and *attribute leakage* (e.g., "yellow mug on a blue book"); GMGM grounds relations (via $\mathcal{L}_{grad}$) and keeps attribute boundaries sharp (via $\mathcal{L}_{feat}$), consistent with its Winoground and POPE gains, and the trend holds under FLUX.2 with fixed seeds.

5 Conclusion and Limitations

We presented **GMGM**, an active synthesis framework that closes the *informativeness gap* of passive text-to-image pipelines by jointly enforcing semantic alignment, SAE-space feature coverage, and gradient informativeness on per-prompt latent codes. A three-term bound links each loss to a distinct source of synthetic-to-real risk, and experiments across six benchmarks and four VLM backbones confirm the predicted gains: GMGM matches the full-budget performance of our strong synthetic baseline (TreeSynth) with only **10%** of the data, improves compositional reasoning ($+58.1\%$ Winoground) and hallucination robustness ($+5.6\%$ POPE F1), and reframes synthetic data as an *optimization* target for closing the data-efficiency bottleneck of multi-modal learning. Its principal limitations are the per-prompt latent-optimization overhead and a reliance on the frozen critic and a small real reference batch, which upper-bound synthetic quality by the critic's blind spots; we discuss these, together with practical mitigations such as cheaper operating points, cross-critic transfer, and modular correction, in Appendix I.

AI USE STATEMENT

AI assistants (OpenAI Codex) were used only for LaTeX formatting, proofreading, and language editing; they did not generate research ideas, experiments, data, or results. The authors verified all AI-assisted edits and take full responsibility for the paper.

ETHICS STATEMENT

GMGM synthesizes training images for MLLM fine-tuning; it does not train or release a generative model, but optimizes latent codes of a *frozen*, publicly available text-to-image generator. All data are established public research datasets (ImageNet-21k, a LAION-2B subset, LLaVA-Instruct-150K, and TreeSynth prompts) used under their licenses; the study involves no human subjects and no personally identifiable information. Because GMGM inherits its frozen critic and generator, synthetic data may reflect their biases and blind spots (e.g., counting, spatial relations, and OCR; Appendix I). As text-to-image generators can be misused to produce misleading media, we recommend standard provenance and content-safety safeguards for any downstream use.

REPRODUCIBILITY STATEMENT

Section 3 specifies the full GMGM procedure and objective, with pseudocode in Algorithm 1, and Appendix B gives all assumptions and complete proofs. Dataset and benchmark construction, the fair-baseline and generator-confound protocols, and the repeated-seed statistical analysis appear in Appendices D–F; implementation details (critic, SAE, proxy probe, loss weights, latent schedule, LoRA recipe, and hardware) are in Section 4.1. We plan to release our code, evaluation scripts, prompt lists, and optimized latent configurations.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4750–4759, 2022.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pp. 370–387. Springer, 2024.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

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. *arXiv preprint arXiv:2306.13394*, 2023.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13869–13920, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

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*, pp. 216–233. Springer, 2024.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. *arXiv preprint arXiv:2504.02821*, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Jiao Sun, Deqing Fu, Yushi Hu, Su Wang, Royi Rassin, Da-Cheng Juan, Dana Alon, Charles Herrmann, Sjoerd Van Steenkiste, Ranjay Krishna, et al. Dreamsync: Aligning text-to-image generation with image understanding feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5920–5945, 2025.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5238–5248, 2022.

Sheng Wang, Pengan Chen, Jingqi Zhou, Qintong Li, Jingwei Dong, Jiahui Gao, Boyang Xue, Jiyue Jiang, Lingpeng Kong, and Chuan Wu. Treesynth: Synthesizing diverse data from scratch via tree-guided subspace partitioning. *arXiv preprint arXiv:2503.17195*, 2025.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36: 73582–73603, 2023.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, et al. Instruction tuning for large language models: A survey. *ACM Computing Surveys*, 58(7):1–36, 2026.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.