G^{2}TR: Generation-Guided Visual Token Reduction for Separate-Encoder Unified Multimodal Models

Junxian Li[1], **Kai Liu**[1], **Zizhong Ding**[1], **Zhixin Wang**[2],
Zhikai Chen[2], **Renjing Pei**[2], **Yulun Zhang**[1*]
[1]Shanghai Jiao Tong University, [2]Huawei Technologies Ltd
lijunxian0531@sjtu.edu.cn

Abstract

The development of separate-encoder Unified multimodal models (UMMs) comes with a rapidly growing inference cost due to dense visual token processing. In this paper, we focus on understanding-side visual token reduction for improving the efficiency of separate-encoder UMMs. While this topic has been widely studied for MLLMs, existing methods typically rely on attention scores, text-image similarity and so on, implicitly assuming that the final objective is discriminative reasoning. This assumption does not hold for UMMs, where understanding-side visual tokens must also preserve the model's capabilities for editing images. We propose G^{2}TR, a generation-guided visual token reduction framework for separate-encoder UMMs. Our key insight is that the generation branch provides a task-agnostic signal for identifying understanding-side visual tokens that are not only semantically relevant but also important for latent-space image reconstruction and generation. G^{2}TR estimates token importance from consistency with VAE latent, performs balanced token selection, and merges redundant tokens into retained representatives to reduce information loss. The method is training-free, plug-and-play, and applied only after the understanding encoding stage, making it compatible with existing UMM inference pipelines. Experiments on image understanding and editing benchmarks show that G^{2}TR substantially reduces visual tokens and prefill computation by **1.94**$\times$ while maintaining both reasoning accuracy and editing quality, outperforming baselines on almost all benchmarks. Code is at `https://github.com/lijunxian111/G2TR`.

1 Introduction

Unified Multimodal Models (UMMs) integrate the capabilities of Multimodal Large Language Models (MLLMs) and image generation models like diffusion or flux models [10, 23]. It can understand the tasks better with rich world knowledge from MLLMs, and then overcome complex image understanding, generation and editing tasks [16, 45, 22, 11]. Modern UMMs can be categorized into two kinds based on their encoders: unified-encoder ones [28, 38] (using one encoder for understanding and generation) or separate-encoder ones [7, 34, 30] (using decoupled encoders for them). Prior work [34] suggests that decoupling visual encoding into separate pathways reduces the conflict between the visual encoder's roles in understanding and generation. Therefore, scaling up and improving capabilities of separate-encoder UMMs have been largely explored.

Given this, a fundamental challenge in UMMs, inference efficiency, is quite emergent for utilizing them. UMMs are typically built with billions, or even tens of billions of parameters, which leads to slow inference and substantial computational cost. This makes real-world deployment difficult, especially in resource-constrained scenarios such as edge devices or latency-sensitive interactive applications. Therefore, improving the inference efficiency of UMMs is increasingly important.

*Corresponding author: Yulun Zhang, yulun100@gmail.com

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

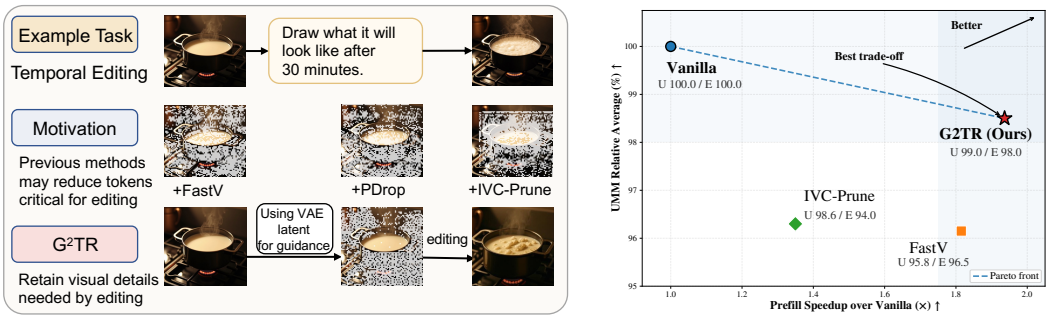

Figure 1: Intuitive view of G^{2}TR. Left: Our method preserves visual tokens crucial for image editing. Right: G^{2}TR lies on the Pareto frontier, compared with previous methods. Notably, UMM relative averages are calculated by: (understanding relative averages + editing relative averages)/2. "U" means understanding relative averages and "E" means editing relative averages (Table 2 and Table 4).

Existing efforts [20, 9] mainly focus on modifying the internal architecture of UMMs, which often introduces considerable engineering complexity. In contrast, visual token reduction offers a simple and training-free alternative yet rarely explored. This naturally motivates our research question:

Can we employ visual token reduction on UMMs to improve the efficiency of them?

Previous works [27, 2, 26, 40, 41] have confirmed the availability of visual token reduction in MLLMs. Regarding them, we simplify the research problem in this paper to reduce understanding-side visual tokens. Thus, we mainly focus on image understanding and editing [7]. The mentioned methods, however, are designed for MLLMs, where visual tokens are selected for textual reasoning, e.g., by attention scores or image-text similarity. This assumption does not fully fit UMMs, since understanding-side visual tokens also condition image editing. **Tokens that seem to have lower attention or text-similarity scores may still preserve visual cues needed for high-quality editing.** Additionally, our observation is that once the pruning ratio is fixed, different pruning rules lead to only limited differences on understanding tasks. Therefore, instead of pursuing a more complex text-centric method, we focus on reducing visual tokens while preserving generation-related information.

Regarding the observations, it becomes our main goal to seek for an effective guidance signal preserving both two mentioned capabilities. Fortunately, the generation branch in UMMs naturally provides such a signal for token importance estimation. During image editing, images are compressed into latent representations that preserve generation-related information. Motivated by this, we argue that understanding-side visual tokens more aligned with these latent representations are more likely to contain visual details critical for image editing. Based on this insight, we propose **G**eneration-**G**uided Visual **T**oken **R**eduction (G^{2}TR), which leverages latents extracted from the VAE branch to guide visual token reduction on the understanding side. Our pipeline consists of two stages: (1) important token selection and (2) redundant token merging. Considering efficiency, effectiveness, and deployment friendliness, we perform token reduction after the ViT encoder. This design serves as a plug-and-play module that can be easily applied to many separate-encoder UMMs. Consistent with our observations, this strategy may not introduce adverse effects on understanding performance.

Extensive experiments are conducted on two separate-encoder UMMs, BAGEL-7B-MoT [7] and InternVL-U [30]. With only 50% visual tokens, G^{2}TR preserves strong performance on both editing and understanding, and achieves the best relative average among token reduction methods, as Figure 1 shows. On BAGEL-7B-MoT, it further reduces the KV cache by **1.90**$\times$ and prefill FLOPs by **1.94**$\times$, without adding extra decoding overhead. Qualitative results also indicate that G^{2}TR keeps critical regions for understanding tasks and preserves local visual details needed for editing. These results suggest that G^{2}TR is a practical way to improve UMM inference efficiency while maintaining both understanding and editing capabilities. We summarize our contributions as follows:

- We propose **G^{2}TR**, a training-free and plug-and-play method for understanding-side visual token reduction for UMMs. To our best of knowledge, we first explore UMMs' visual token reduction.
- We introduce and analyze the novel **generation-guided strategy**, which includes multi-stage importance estimation and token merging. This insight can be extended to other UMMs.
- Extensive experiments suggest that **G^{2}TR** can both preserve understanding and generation performance, while improve the efficiency of separate-encoder UMMs as well.

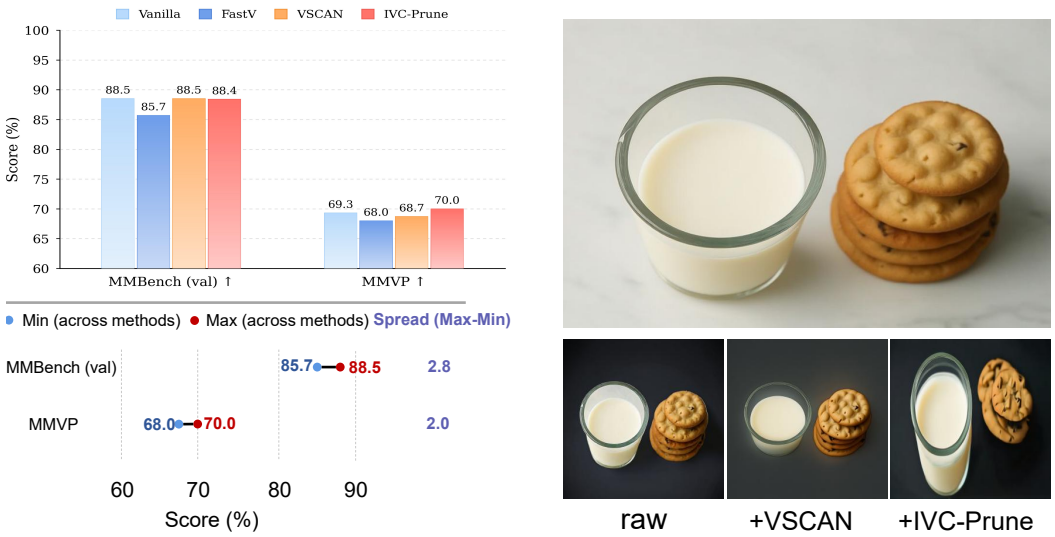

Figure 2: The two observations under this scenario. (1) Left figure indicates that limited difference exists in understanding tasks under different visual token reduction methods. (2) Right figure shows the results under prompt "Edit this image: change the background to be darker". Using previous visual token reduction methods cause worse editing results.

2 Related Work

UMMs & separate-encoder UMMs. UMMs integrate understanding capabilities of MLLMs [17, 32, 29, 46, 43, 44, 12, 14] and generation capabilities of diffusion or flow models [5, 23, 42, 10]. Recent works on this topic [1, 37, 3, 7, 15, 30, 13, 11] try to develop end-to-end UMMs based on this idea, providing strong multimodal reasoning capabilities benefiting generation. The latest state-of-the-art models, like GPT-Image-1 [21] and Gemini3-Pro-Image [4] see a huge improvement in understanding detailed and complex instructions and generating images. Many recent open-source UMMs [3, 7, 15] choose a separate-encoder architecture, and serve as the focus of this work.

Efficient UMMs. Currently, efficient UMM is a research field still largely underexplored. Several existing methods focus on techniques modifying architectures, like Mao et al. [20] proposed a mixture-of-depth method to solve tasks of different difficulties. While He et al. [9] also utilizes depth pruning and width reduction on UMMs to obtain efficiency. Such methods work well, while often are more costly or not very easy for real-world utilization. In contrast, visual token reduction, serving as a simple training-free method, are rarely tried for UMMs.

Visual token reduction for multimodal models. Visual token reduction is first developed in MLLMs. Usually it means two techniques: token merging and token pruning [25]. Earlier methods simply utilize attention scores or image-text feature similarities as guidance to reduce visual tokens [2, 26]. Such guidance may fall short in visual redundancy, detailed structure, or task-specific token utility. More recent works [41, 27, 40] utilize techniques like RoPE scores, local-global attention scores and so on to address these issues. Adapting to separate-encoder UMMs, we design unique methods based on generation-branch guidance for better visual token reduction.

3 Preliminaries

3.1 Separate-Encoder Architecture

UMMs aim to support both visual understanding and visual generation within a single architecture. Recent designs [34] suggest that these two capabilities impose different requirements on visual representations: understanding favors semantic and abstract visual features, while generation and editing requires low-level appearance and visual details. (Appendix A better shows the relationship among generation, editing, and understanding-side visual tokens.) Therefore, instead of using a single visual encoder for both tasks, a separate-encoder architecture decomposes this into an understanding branch and a generation branch. Given an input image x, the understanding encoder E_{u}, typically instantiated as a ViT-style vision encoder, maps x into a sequence of N_u semantic visual tokens:

$$\mathbf{U} = E_{\mathrm{u}}(x) = \{\mathbf{u}_i\}_{i=1}^{N_u}, \qquad \mathbf{u}_i \in \mathbb{R}^{d_u}, \tag{1}$$

where $\mathbf{U}$ denotes the sequence and $\mathbf{u}_i$ denotes each understanding-side visual token.

Table 1: Signals for guiding visual token selection of previous methods. Attention score and image-text similarity are often utilized. These designs are not suitable for directly applied in UMMs.

Name	Visual Token Selection Signal
FastV [2]	inner LLM attention score
W-FastV [33]	window mechanism + inner LLM attention score
PDrop [39]	inner LLM attention score
VSCAN [41]	ViT attention score + inner LLM image-text similarity
IVC-Prune [27]	RoPE + image-text similarity

These tokens are projected into hidden state ready for inner structures $\mathbf{H}_{\mathrm{u}}$ by an MLP projector P_{u}:

$$\mathbf{H}_{\mathrm{u}} = P_{\mathrm{u}}(\mathbf{U}) + \mathbf{P}_{\mathrm{u}}^{\mathrm{pos}}, \tag{2}$$

where $\mathbf{P}_{\mathrm{u}}^{\mathrm{pos}}$ denotes visual positional embeddings. This branch mainly preserves high-level semantic cues for recognition, reasoning, and instruction following. In parallel, the generation branch usually employs a VAE-style encoder E_{g} to obtain N_g compact latent tokens:

$$\mathbf{Z} = E_{\mathrm{g}}(x) = \{\mathbf{z}_j\}_{j=1}^{N_g}, \qquad \mathbf{z}_j \in \mathbb{R}^{d_g}, \tag{3}$$

where $\mathbf{Z}$ denotes the sequence and $\mathbf{z}_j$ denotes each latent token. The latent tokens are mapped to the same feature hidden space through another MLP projector P_{g}:

$$\mathbf{H}_{\mathrm{g}} = P_{\mathrm{g}}(\mathbf{Z}) + \mathbf{P}_{\mathrm{g}}^{\mathrm{pos}} + \mathbf{e}(t), \tag{4}$$

where $\mathbf{H}_{\mathrm{g}}$ is the hidden state and $\mathbf{e}(t)$ is a timestep embedding when the model is trained with diffusion [23] or flow-matching [10] objectives. The generation branch thus retains information that is essential for reconstructing or synthesizing images. Finally, the unified inner transformer layers consumes text embeddings $\mathbf{H}_{\mathrm{text}}$ together with tokens from either or both branches. They interact with each other through shared structures (like LLM decoders [7]).

This formulation highlights a useful asymmetry: ViT tokens provide semantically discriminative but often redundant visual evidence, whereas VAE latent tokens provide compact generative signals that preserve image structure. In this work, we build on this separate-encoder paradigm and study how the generation branch can guide the reduction of understanding-side visual tokens without disrupting the unified multimodal interface, thus protecting both image understanding and editing.

3.2 Observations

Observation 1: different visual token reduction methods lead to limited difference on understanding capabilities. Seeing Figure 2 left, we select BAGEL-7B-MoT [7] and some SOTA (state-of-the-art) text-centric visual token reduction methods [2, 41, 27] for MLLMs. Table 1 reports the signals for guiding important visual token selection of the SOTA methods. "Spread" is defined as the maximum - minimum score on a benchmark. Different visual token reduction methods show close performance on understanding benchmarks. On MMBench, the scores range from 85.7 to 88.5, with only a 2.8 spread. On MMVP, the range is also narrow, from 68.0 to 70.0. We also find that no visual token reduction rule shows a clear and consistent advantage. This suggests that developing more complex text-centric methods for UMMs may not be very cost-effective.

Observation 2: Visual tokens chosen by previous methods are not consistent with image editing importance. Figure 2 right reports the visualization results of InternVL-U [30] (with or without token reduction methods) on image editing. We discover that, previous methods can finish the task not so well or damage local visual details, even when the instruction is simple. This shows that text-centric token importance is not enough for UMMs, and this becomes our core research problem.

4 Methodology

4.1 Insight and Method Overview

Regarding Section 3, our key insight is that a ViT token well aligned with its corresponding VAE latent is more likely to carry information that should be kept for both understanding and generation. Therefore, the generation branch can serve as a natural guide for estimating understanding-side visual token importance and reducing the number of them.

Based on this insight, we propose $\mathbf{G}^2\mathbf{TR}$, a training-free visual token reduction method for separate-encoder UMMs. Given an input image, we first extract full visual tokens from the understanding encoder and latent anchors from the generation-side VAE. Each visual token is then scored by its

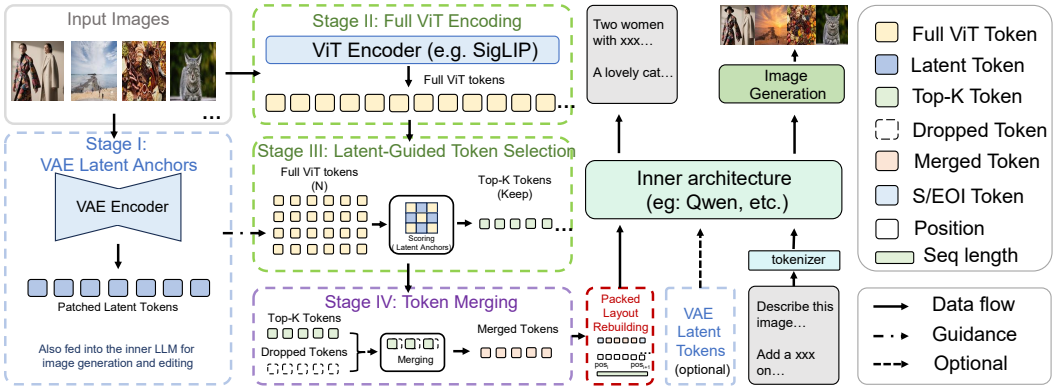

Figure 3: Method Overview. Here we focus on tasks which understanding-side visual tokens are necessary. S/EOI means "start of image" and "end of image". The whole visual token reduction process consists of four stages: firstly the VAE generates latent tokens (also serving as guidance) and the ViT generate full visual tokens; then the top-K important visual tokens are selected based on the guidance strategy; finally, the dropped visual tokens are merged onto nearest important tokens.

consistency with the corresponding latent anchor. To avoid concentrating tokens in a few salient regions, the selection is performed with balanced selection according to latent anchors. The removed tokens are not directly discarded. Instead, they are merged into the most similar retained tokens in feature space, which keeps the token budget low while reducing information loss, according to [41, 24]. G^{2}TR is applied after the ViT encoder and before the inner LLM prefill. This placement keeps the original encoders and attention operators unchanged.

4.2 Latent-Guided Importance Estimation

See Figure 3, Stage III. We use the notations following Sec. 3. Before scoring, we map them into the same hidden dimension. For simplicity, we still use $\mathbf{u}_i$ and $\mathbf{z}_j$ to denote the aligned features.

The sequences of ViT tokens and VAE latent tokens can then be mapped into two 2D grids. For further actions, the VAE latent tokens are specially processed. An operation similar to mean pooling is used, where four adjacent VAE latent tokens on the VAE grid are pooled into one latent anchor. Denote sizes of the ViT grids and the processed VAE grids as $H_u \times W_u$ and $H_g \times W_g$. For a visual token $\mathbf{u}_i$ located at (p_i, q_i) on the ViT grid, we assign it to the corresponding latent anchor by

$$c(i) = \left(\left\lfloor \frac{p_i H_g}{H_u} \right\rfloor, \left\lfloor \frac{q_i W_g}{W_u} \right\rfloor \right), \tag{5}$$

where $\lfloor \rfloor$ denotes the flooring option. We then compute its importance score as the cosine similarity to the matched VAE latent $\mathbf{z}_{c(i)}$:

$$s_i = \cos(\mathbf{u}_i, \mathbf{z}_{c(i)}) = \frac{\mathbf{u}_i^\top \mathbf{z}_{c(i)}}{\|\mathbf{u}_i\|_2 \|\mathbf{z}_{c(i)}\|_2}. \tag{6}$$

A larger score means that the visual token is more consistent with the local generative latent. Such tokens are more likely to preserve visual factors that matter for image editing.

Given a keep ratio ρ, the final token budget is

$$K = \min\left(N_u, \max(K_{\min}, \lfloor \rho N_u \rceil)\right), \tag{7}$$

where $K_{\min}$ is the minimum number of retained tokens and $\lfloor \rceil$ denotes rounding to the nearest integer. Direct top-K selection may concentrate tokens in a few high-score regions. To keep the overall balance, we first select the best token inside each non-empty latent anchor:

$$b_{\mathbf{c}} = \arg\max_{i:c(i)=\mathbf{c}} s_i, \qquad \mathbf{c} \in \{1, \ldots, H_g\} \times \{1, \ldots, W_g\}. \tag{8}$$

This gives a candidate set $\mathcal{B} = \{b_{\mathbf{c}}\}$. We first select the top-K highest-scoring candidates from $\mathcal{B}$:

$$\mathcal{S}_0 = \mathrm{TopK}\left(\mathcal{B}, \min(K, |\mathcal{B}|)\right). \tag{9}$$

If $|\mathcal{S}_0| < K$, we fill the remaining budget with highest-scoring tokens from the unselected set only once:

$$\mathcal{S} = \mathcal{S}_0 \cup \mathrm{TopK}\left(\{1, \ldots, N_u\} \setminus \mathcal{S}_0, K - |\mathcal{S}_0|\right). \tag{10}$$

Otherwise, we set $\mathcal{S} = \mathcal{S}_0$. The resulting set $\mathcal{S}$ contains the retained visual tokens. It favors tokens aligned with generation-side latent while avoiding concentration in a few local regions.

4.3 Redundant Token Merging

Seeing Figure 3, Stage IV, after obtaining the retained token set $\mathcal{S}$, a simple solution is to discard all redundant tokens calculated in the selection process. However, direct pruning may remove fine-grained visual cues that are not selected by the importance score. To reduce information loss, we merge the removed tokens into the retained ones in feature space.

Let $\bar{\mathcal{S}} = \{1, \ldots, N_u\} \setminus \mathcal{S}$ denote the redundant token set. For each removed token $\mathbf{u}_i$ in $\bar{\mathcal{S}}$, we find its nearest retained token by cosine similarity:

$$n(i) = \arg\max_{j \in \mathcal{S}} \cos(\mathbf{u}_i, \mathbf{u}_j), \quad i \in \bar{\mathcal{S}}. \tag{11}$$

Each retained token then absorbs the removed tokens assigned to it:

$$\tilde{\mathbf{u}}_j = \frac{\lambda \mathbf{u}_j + \sum_{i \in \bar{\mathcal{S}}: n(i)=j} \mathbf{u}_i}{\lambda + \left|\{i \in \bar{\mathcal{S}} : n(i) = j\}\right|}, \quad j \in \mathcal{S}. \tag{12}$$

Here, λ controls the weight of the original retained token. Following works like [41], we set λ just to 1.0. Finally, the final compressed visual sequence is

$$\tilde{\mathbf{U}} = \{\tilde{\mathbf{u}}_j \mid j \in \mathcal{S}\}. \tag{13}$$

We keep the order of the retained indices when forming $\tilde{\mathbf{U}}$. This makes the output sequence shorter while preserving part of the information from removed tokens. With our design, the inner architectures of UMMs receive a much shorter sequence of understanding-side tokens.

4.4 Layout Rebuilding

After merging, the shortened visual tokens usually no longer match the original `<IMG_CONTEXT>` span. This alignment is important for UMMs, since the inner LLM hidden states also condition image editing. We therefore rebuild the input before the inner LLM prefill: the original understanding-level visual tokens are replaced by the merged tokens $\tilde{\mathbf{U}}$, while text tokens and image boundary tokens are kept unchanged. The input ids, attention mask, and position ids are updated and padded across the batch. This keeps the inner UMM architecture unchanged, while letting both the inner LLM and the generation decoder work on the same shortened sequence.

4.5 Flash-Attention Compatibility

Modern large models are often accelerated through Flash-Attention [6]. Given this, a practical advantage of our G^{2}TR is its strong compatibility with this acceleration support. The implementation of existing methods, such as VScan-like [41] variants, often need explicit attention maps to estimate token importance, or introduce pruning inside the language model. This may require extra standard attention score computation and complicate KV-cache management under Flash-Attention. This issue happens in common implementations, where Flash-Attention is used for the forward pass but attention weights are separately reconstructed by dense matmul and softmax for token scoring. In contrast, our method performs token reduction only before inner prefilling. The inner architecture therefore only receives a shorter sequence, without changing the attention operator or requesting attention weights. This plug-and-play design for separate-encoder UMMs keeps the original Flash-Attention kernel unchanged while directly reducing computational cost.

5 Experiments

5.1 Experiment Settings

The purpose of our experiments is to answer two questions: (1) can our method keep understanding and editing performance; (2) can our method really improve efficiency of UMMS.

Models. We mainly choose one famous separate-encoder UMM, BAGEL-7B-MoT (BAGEL-7B) [7]. A very recent separate-encoder UMM, InternVL-U (4B) [30], is also employed. The evaluation codes are based on their official repository. Scores of InternVL-U are reproduced on our own.

Benchmarks. We evaluate the performances on tasks where understanding-side visual tokens are necessary: image understanding and image editing. For image understanding, we choose MME [8], MMBench [19], MMVP [31] and RealWorldQA (RWQA) [36]. Results of BAGEL-7B on MMVP are implemented by ourselves. For image editing, we adopt GEdit-Bench (GEdit, only evaluating English queries) [18], IntelligentBench (Intell.) [7], and RISE-Bench (RISE) [45].



Input | Vanilla | FastV [2] | PDrop [39] | IVC-Prune [27] | **G^{2}TR (ours)**

Figure 4: Visualization of Editing results. The instructions of the four groups of images from top to down are: "Draw what it will look like after being frozen."; "Draw what it will look like after one hour on a hot grill."; "What does this dish look like after it has been baked?"; "Can I see the appearance of this ring on a finger?". Generally, G^{2}TR performs better in UMMs for image editing than compared baselines.

Baselines. Following IVC-prune [27] (ICLR'26), we choose some SOTA visual token reduction methods from top-tier venues: FastV [2], Window FastV (W-FastV) [33], PDrop [39], VSCAN [41] and IVC-prune [27] itself. We use the official implementation method and layers for visual token reduction from their official repository. "Vanilla" means the original performance of utilized UMMs.

Other settings. We test all the models, benchmarks and baselines on one A6000 48G GPU once. The default generation hyper-parameters are chosen. Details of utilized models, benchmarks, and hyper-parameters are in Appendix C.

5.2 Main Performance

Performance on image editing tasks. Figure 4 reveals some image editing results of G^{2}TR and baselines. Images and instructions are from RISE and Intell. Compared with the original results, several baselines produce blurred details or fail to follow the editing instructions. In more severe cases, the edited images deviate greatly from the input image, such as the result of FastV in the first group and the result of IVC-prune in the third group. In contrast, after applying G^{2}TR, the model can largely follow the instructions and complete the edits successfully. It also preserves an editing quality comparable to the vanilla model setting. This supports the core claim in Section 4.1.

Performance on image understanding tasks. Table 2 reports the results on image understanding benchmarks. G^{2}TR achieves the best relative average among all baselines on both UMMs, reaching 94.9% on InternVL-U and 99.0% on BAGEL-7B while keeping only 50% visual tokens. On BAGEL-7B, G^{2}TR matches the vanilla performance on MMBench, obtains the best result on MMVP, and gives the smallest drop on RealWorldQA. On InternVL-U, it also achieves the best or tied-best results on three benchmarks among all compared baselines. These results show that our reduction does not sacrifice understanding ability. Instead, the VAE latent guidance may help retain several visual tokens that are useful for both generation and understanding. This may be because the generation and understanding branches in UMMs can provide complementary signals. Generally, performances on image understanding and editing provide positive answers to question (1) in Section 5.1.

Efficiency. Table 3 reports efficiency of our method compared with baselines. G^{2}TR reduces the average visual tokens to 50%, leading to a 1.90× KV-cache reduction and a 1.94× prefill FLOPs speedup over the vanilla model. Compared with FastV, G^{2}TR further lowers the FLOPs ratio from 55.10% to 51.64%, while using fewer tokens. It also clearly outperforms IVC-Prune under the same token budget, reducing prefill FLOPs from 3.0619T to 2.1341T. The decode latency is slightly

Table 2: Performance of G^{2}TR compared with baselines on several understanding benchmarks. "Avg." means average and "Rel. Avg." means relative average. We **highlight** the best performances and underline the second best ones. Generally, G^{2}TR reaches the best relative average across both UMMs.

Models	Methods	Avg. Tokens	MME	MMBench (dev)	MMVP	RWQA	Rel. Avg.
InternVL-U	Vanilla	100%	2057	80.5	46.0	48.8	100%
	FastV [2][ECCV'24]	54%	**2037**	77.9	38.0	45.5	92.9%
	W-FastV [33][ACL'25]	54%	1926	78.6	39.3	45.2	92.3%
	PDrop [39][CVPR'25]	61%	1953	77.7	38.7	45.6	92.0%
	VSCAN [41][TMLR'26]	50%	1968	78.1	**40.0**	43.5	92.2%
	IVC-Prune [27][ICLR'26]	50%	2020	78.4	39.0	46.0	93.7%
	G^{2}TR (ours)	50%	2034	**78.8**	**40.0**	**46.8**	**94.9%**
BAGEL-7B	Vanilla	100%	2388	88.5	69.3	60.2	100%
	FastV [2][ECCV'24]	54%	2306	85.7	68.0	55.2	95.8%
	W-FastV [33][ACL'25]	54%	2287	85.9	67.3	55.4	95.6%
	PDrop [39][CVPR'25]	61%	2312	84.0	66.7	53.8	94.1%
	VSCAN [41][TMLR'26]	50%	**2339**	**88.5**	68.7	48.4	94.4%
	IVC-Prune [27][ICLR'26]	50%	2332	88.4	**70.0**	57.8	98.6%
	G^{2}TR (ours)	50%	2318	**88.5**	**70.0**	**59.0**	**99.0%**

Table 3: Efficiency comparison on BAGEL-7B-MoT. Size of KV Cache, average visual tokens (Avg. Tokens), FLOPs ratio, prefill FLOPs and decode latency are evaluated.

Model	Method	KV Cache (MB) ↓	Avg. Tokens (%) ↓	FLOPs Ratio ↓	Prefill FLOPs (T) ↓	Decode Latency (ms/token) ↓
BAGEL-7B	Vanilla	42.02 (1.00×)	100% (1.00×)	100.00% (1.00×)	4.1325 (1.00×)	49.60 (1.00×)
	FastV	**22.04 (1.91×)**	54% (1.85×)	55.10 % (1.85×)	2.2769 (1.82×)	48.04 (1.03×)
	IVC-Prune	22.06 (1.90×)	**50% (2.00×)**	74.09% (1.37×)	3.0619 (1.35×)	49.30 (1.01×)
	G^{2}TR (Ours)	22.06 (1.90×)	**50% (2.00×)**	**51.64% (1.94×)**	**2.1341 (1.94×)**	**47.88 (1.04×)**

improved to 47.88 ms/token, showing that our method brings consistent efficiency gains without introducing extra decoding overhead, answering question (2) in Section 5.1.

Quantitative results of image editing. Beyond the direct visual comparisons, we further measure image editing quality with the original evaluation protocols of utilized benchmarks. As shown in Table 4, G^{2}TR also shows clear numerical advantages over existing token reduction baselines. On GEdit-Bench (EN), our method achieves the best results among compressed variants on all three metrics. It also obtains 44.0 on IntelligentBench, close to the vanilla model while using only half of the visual tokens. Overall, G^{2}TR reaches a 98.0% relative average, outperforming other baselines.

Thinking mode. We evaluate the performance when opening the thinking mode of BAGEL-7B-MoT. Following their original report [7], we evaluate RISE [45] only when thinking mode is open here. For understanding benchmarks, we set the max response token length to 2048 for avoiding truncation. Table 5 reports the results. We observe no consistent advantage of thinking mode over the non-thinking setting across benchmarks. Under this setting, G^{2}TR still shows a stable trade-off. It reaches the best relative average among all token reduction methods, preserving 95.6% of the vanilla performance. It also matches the vanilla model on RealworldQA and remains close on IntelligentBench, while keeping competitive results on MMVP. These results indicate that G^{2}TR is still effective when UMMs use thinking mode, and does not rely on a specific decoding style.

Keep ratio. Table 6 studies the performance under different keep ratios. G^{2}TR consistently achieves the best or tied-best results across three token budgets on both MMVP and IntelligentBench. Notably, its advantage remains clear under more aggressive reduction: at a 25% keep ratio, G^{2}TR reaches 60.0 on MMVP and 42.0 on IntelligentBench, outperforming both FastV and IVC-Prune. These results demonstrate that G^{2}TR remains robust across different levels of visual token reduction.

5.3 Ablation study.

To study whether our guidance source is the better choice compared with others, we make this study. In detail, we also employ three selection signals: random, attention-based importance and image-text feature similarity (text similarity) right after the ViT encoding process. Table 7 reports the results. It can be discovered that using VAE latent as guidance achieves best scores across both understanding and

Table 7: Ablation on guidance sources. We **highlight** the best performances.

Guidance Source	MMVP ↑	RWQA ↑	Intell. ↑
Random	65.3	48.2	40.6
Attention-based	68.7	55.8	42.6
Text similarity	68.0	55.2	41.7
VAE latent (ours)	**70.0**	**59.0**	**44.0**

Table 4: Quantitative results on image editing. We **highlight** and underline the best and second best performances.

Models	Methods	GEdit (EN) G_SC	G_PQ	G_O	Intell.	Rel. Avg.
BAGEL-7B	Vanilla	7.36	6.83	6.52	44.9	100%
	FastV	7.09	6.58	6.35	43.0	96.5%
	VSCAN	7.03	6.65	6.32	42.6	96.2%
	IVC-Prune	6.82	6.52	6.27	41.1	94.0%
	G^{2}TR (ours)	**7.18**	**6.67**	**6.42**	**44.0**	**98.0%**

Table 5: Results when opening thinking mode. We **highlight** and underline the best and second best performances.

Models	Methods	MMVP	RWQA	RISE	Intell.	Rel. Avg.
BAGEL-7B w/ CoT	Vanilla	64.7	63.4	11.9	55.3	100%
	FastV	63.3	62.2	9.1	51.9	91.6%
	VSCAN	**66.7**	60.2	8.3	53.7	91.2%
	IVC-Prune	64.0	62.1	7.6	51.4	88.4%
	G^{2}TR (ours)	66.0	**63.4**	**9.8**	**54.3**	**95.6%**

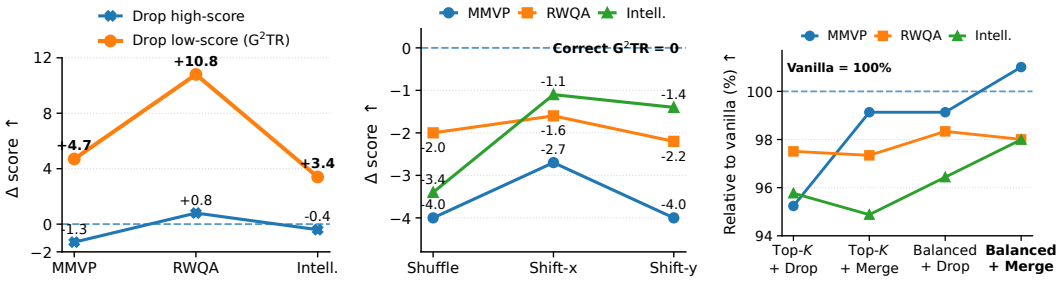

Figure 5: Further analysis. (a) Validity of the VAE-alignment score. (b) Effect of perturbing the spatial token-to-anchor relationship. (c) Component ablation of balanced selection and token merging.

editing tasks. This ablation supports our claim that our proposed guidance plays a core role in the good performance above, compared with other selection signals.

5.4 Further Analysis

We further analyze G^{2}TR from three aspects: score validity, spatial correspondence, and component design. Results are in Figure 5. First, removing low-score tokens consistently outperforms random removal by 4.7, 10.8, and 3.4 points on MMVP, RWQA, and IntelligentBench, respectively, whereas removing high-score tokens provides no consistent gain over random removal. This supports the importance ranking induced by VAE alignment. Second, perturbing the token-to-anchor correspondence through shuffling or spatial shifts consistently degrades performance relative to the original assignment, showing that the gains cannot be explained by arbitrary matching with VAE anchors. Finally, balanced selection consistently improves direct dropping over global top-K selection. Token merging has benchmark-dependent effects when used alone, but together with balanced selection it yields the strongest overall performance with improving performances of two benchmarks. These results further validate the spatial correspondence and individual reduction components of G^{2}TR.

5.5 Visualization of retained tokens.

We further do this to show what tokens are kept after reduction. As shown in Figure 6, the retained tokens preserve visual information useful for both understanding and editing. In the counting example, the retained tokens still cover the pedestrians and allow the model to predict the correct answer. In the editing example, our method preserves most visual tokens of the pot and also visual tokens for the whole layout. The details are important for producing a well-edited image. This suggests that G^{2}TR can keep tokens not only effective for textual understanding, but also friendly to image editing.

6 Limitations and Future Work

This work focuses on visual token reduction for separate-encoder UMMs, where the generation branch can provide a natural guidance signal. As UMM architectures continue to evolve, future models may adopt more native and adaptive interactions between understanding and generation pathways. Extending generation-guided reduction to these broader designs is an interesting future direction, which is not explored in our paper. In addition, this paper studies a fixed token budget for clear comparison. A more dynamic strategy that adjusts the reduction ratio based on task difficulty, image content, or latency constraints may further improve the trade-off between efficiency and performance. We believe such directions will help make UMMs more practical for interactive and resource-constrained applications.

Table 6: Results on different keep ratios. We **highlight** the best performances under the same keep ratio. G^{2}TR keeps the good performance under ratios of 50%, 37.5% and 25%.

Benchmarks	50%			37.5%			25%		
	FastV	IVC-Prune	**G^{2}TR (ours)**	FastV	IVC-Prune	**G^{2}TR (ours)**	FastV	IVC-Prune	**G^{2}TR (ours)**
MMVP	68.0	**70.0**	**70.0**	60.3	58.7	**63.3**	54.7	56.0	**60.0**
Intell.	43.0	41.1	**44.0**	41.8	41.0	**42.5**	40.3	41.0	**42.0**

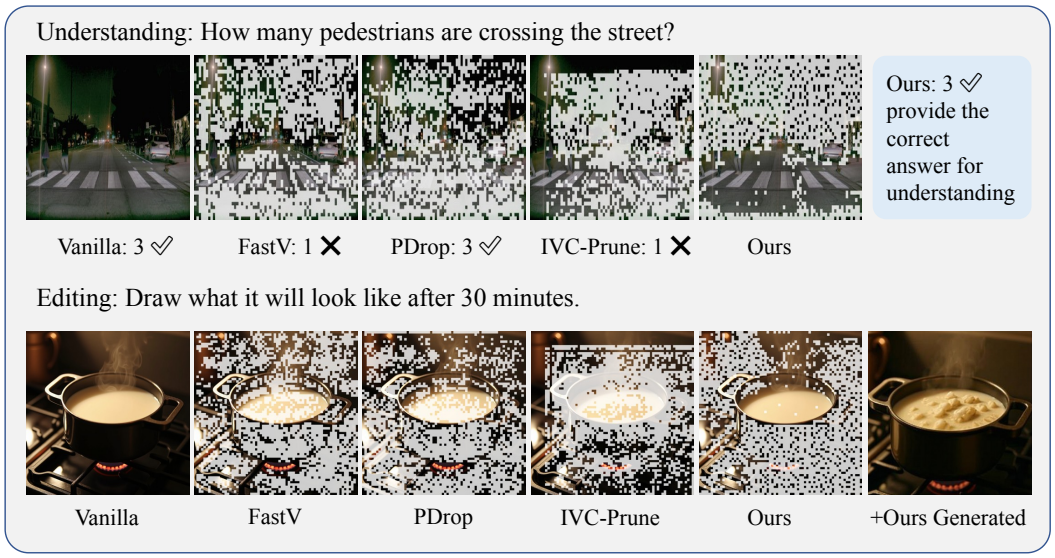

Figure 6: Visualization of retained tokens. The first line illustratively shows one case of image understanding, and the second line shows one case of image editing.

7 Conclusion

In this work, we propose G^{2}TR, a training-free and plug-and-play framework that uses the generation branch as guidance for reducing understanding-side visual tokens for UMMs. Different from previous works mainly using text-centric guidance, G^{2}TR preserves visual tokens that are more consistent with generation-related latent representations. Our design reduces the token budget before inner LLM prefill while keeping the attention operators unchanged. Extensive experiments show that G^{2}TR improves inference efficiency and maintains strong performance on both understanding and image editing tasks. These results suggest that G^{2}TR is a simple and effective direction for efficient UMM inference, thereby benefiting UMM deployment and utilization under resource-constrained scenarios. We hope that G^{2}TR can facilitate the new research trends on UMM efficiency.

Acknowledgements

This work was supported by Huawei Explore X funds. This work was also sponsored by Al Hundred Schools Program and was carried out using the Ascend AI technology stack.

Competing interests. The authors declare no additional competing interests beyond the affiliations and funding disclosed above.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 2024.

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

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

[5] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *TPAMI*, 2023.

[6] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022.

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

[8] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *NeurIPS*, 2025.

[9] Shwai He, Chaorui Deng, Ang Li, and Shen Yan. Understanding and harnessing sparsity in unified multimodal models. *arXiv preprint arXiv:2512.02351*, 2025.

[10] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[11] Junxian Li, Kai Liu, Leyang Chen, Weida Wang, Zhixin Wang, Jiaqi Xu, Fan Li, Renjing Pei, Linghe Kong, and Yulun Zhang. Planviz: Evaluating planning-oriented image generation and editing for computer-use tasks. *arXiv preprint arXiv:2602.06663*, 2026.

[12] Junxian Li, Di Zhang, Xunzhi Wang, Zeying Hao, Jingdi Lei, Qian Tan, Cai Zhou, Wei Liu, Yaotian Yang, Xinrui Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Wei Li, Mao Su, Shufei Zhang, Wanli Ouyang, Yuqiang Li, and Dongzhan Zhou. Chemvlm: Exploring the power of multimodal large language models in chemistry area. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 415–423, 2025.

[13] Qiming Li, Xiaocheng Feng, Yixuan Ma, Ruihan Chen, Zihe Tong, Zekai Ye, Xiachong Feng, Libo Qin, Haoyu Ren, Kun Chen, Yunfei Lu, Dandan Tu, and Bing Qin. Unlocking multilingual reasoning capability of llms and lvlms through representation engineering. In *ACL*, 2026.

[14] Sichao Li, Sai Ma, Daniel Kilov, Secil Yanik Guyot, Zhuang Li, and Seth Lazar. Nora: Evaluating grounded reasonableness in visual first-person normative action reasoning. *arXiv preprint arXiv:2606.04806*, 2026.

[15] Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *CVPR*, 2025.

[16] Yongyuan Liang, Wei Chow, Feng Li, Ziqiao Ma, Xiyao Wang, Jiageng Mao, Jiuhai Chen, Jiatao Gu, Yue Wang, and Furong Huang. Rover: Benchmarking reciprocal cross-modal reasoning for omnimodal generation. In *ICLR*, 2026.

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[18] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[19] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, 2024.

[20] Weijia Mao, Zhenheng Yang, and Mike Zheng Shou. Unimod: Efficient unified multimodal transformers with mixture-of-depths. *arXiv preprint arXiv:2502.06474*, 2025.

[21] OpenAI. Introducing our latest image generation model in the api. `https://openai.com/index/image-generation-api/`, 2025.

[22] Kaihang Pan, Weile Chen, Haiyi Qiu, Qifan Yu, Wendong Bu, Zehan Wang, Yun Zhu, Juncheng Li, and Siliang Tang. Wiseedit: Benchmarking cognition- and creativity-informed image editing. In *CVPR*, 2026.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[24] Kele Shao, TAO Keda, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. In *NeurIPS*, 2025.

[25] Kele Shao, TAO Keda, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. A survey of token compression for efficient multimodal large language models. *TMLR*, 2026.

[26] Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael X Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal llms. In *COLING*, 2025.

[27] Zhichao Sun, Yidong Ma, Gang Liu, Yibo Chen, Xu Tang, Yao Hu, and Yongchao Xu. Ivc-prune: Revealing the implicit visual coordinates in lvlms for vision token pruning. In *ICLR*, 2026.

[28] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[29] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[30] Changyao Tian, Danni Yang, Guanzhou Chen, Erfei Cui, Zhaokai Wang, Yuchen Duan, Penghao Yin, Sitao Chen, Ganlin Yang, Mingxin Liu, et al. Internvl-u: Democratizing unified multimodal models for understanding, reasoning, generation and editing. *arXiv preprint arXiv:2603.09877*, 2026.

[31] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024.

[32] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[33] Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In *ACL Findings*, 2025.

[34] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *CVPR*, 2025.

[35] Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. Kris-bench: Benchmarking next-level intelligent image editing models. In *NeurIPS*, 2025.

[36] xAI. Announcing grok-1.5. `https://x.ai/news/grok-1.5`, 2024.

[37] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *ICLR*, 2025.

[38] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *NeurIPS*, 2025.

[39] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *CVPR*, 2025.

[40] Rui Xu, Yunke Wang, Yong Luo, and Bo Du. Rethinking visual token reduction in lvlms under cross-modal misalignment. In *AAAI*, 2026.

[41] Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. Vscan: Rethinking visual token reduction for efficient large vision-language models. *TMLR*, 2026.

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[43] Yu Zhang, Jinlong Ma, Yongshuai Hou, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. Evaluating and steering modality preferences in multi-modal llms. In *ICML*, 2026.

[44] Yu Zhang, Mufan Xu, Xuefeng Bai, Kehai Chen, Pengfei Zhang, Yang Xiang, and Min Zhang. Instruction anchor: Dissecting the mechanistic dynamics of modality arbitration. *arXiv preprint arXiv:2602.03677*, 2026.

[45] Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, Hua Yang, Xue Yang, and Haodong Duan. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. In *NeurIPS*, 2025.

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

A Relationship among Generation, Editing and Understanding-side Visual Tokens

Image generation (here "generation" is a task name) and image editing are two sub-tasks in UMMs. Image generation creates an image from an instruction without a given image. Image editing takes a source image and an instruction, and modifies the source image according to the instruction. Both tasks finally rely on the generation branch (here "generation" means a branch name) to produce image results. The key difference lies in the input path. Pure image generation does not involve understanding-side visual tokens, since no source image is encoded by the vision encoder. In contrast, image editing also represents the source image as understanding-side visual tokens, which are then used to condition the generation process.

Therefore, among these two tasks, only image editing is directly affected by understanding-side visual token reduction. For simplicity, when we mention "visual tokens", we are talking about understanding-side visual tokens.

B Notations in this paper

We list all the notations in our main paper in Table 8.

Table 8: Notations used in our paper

Notations	Meanings
x	input image
E_{u}	understanding encoder (usually ViT)
$\mathbf{U}$	the understanding-side visual token sequence
N_u	the number of understanding-side visual tokens
$\mathbf{u}_i$	each visual token
$\mathbf{H}_{\mathrm{u}},\ \mathbf{H}_{\mathrm{g}}$	hidden states from understanding branch and generation branch
$P_{\mathrm{u}},\ P_{\mathrm{g}}$	MLP projectors of the two branches
$\mathbf{P}_{\mathrm{u}}^{\mathrm{pos}}$	understanding visual position embeddings
E_{g}	generation encoder (usually VAE)
$\mathbf{Z}$	the generation-side latent token sequence
N_g	the number of generation-side latent tokens
$\mathbf{z}_j$	each latent token
$\mathbf{P}_{\mathrm{g}}^{\mathrm{pos}}$	generation-side position embeddings
$\mathbf{e}(t)$	time embeddings
$\mathbf{H}_{\mathrm{text}}$	hidden states from text instructions
(p_i, q_i)	the position of a visual token
$c(i)$	the position of a latent anchor corresponding to a visual token
$H_u,\ W_u,\ H_g,\ W_g$	the height and width of grids
$\lfloor\rfloor$	floor option
$\mathbf{z}_{c(i)}$	matched VAE latent
ρ	keep ratio
K	token budget
s_i	importance score
$b_{\mathbf{c}}$	the best token inside each non-empty latent anchor
$\mathcal{B}$	the candidate set
$\mathcal{S},\ \mathcal{S}_0$	retained tokens
$\bar{\mathcal{S}}$	redundant tokens
$n(i)$	index of the retained token nearest to removed token $\mathbf{u}_i$
λ	merging weight
$\tilde{\mathbf{u}}_j$	final retained tokens
$\tilde{\mathbf{U}}$	final compressed visual token sequence

C Detailed Experiment Settings

C.1 Model Usage

BAGEL. BAGEL [7] is an open-source UMM that supports both multimodal understanding and visual generation. It uses a decoder-only design and is pretrained on large-scale interleaved text, image, video, and web data. The released BAGEL-7B-MoT model has 7B active parameters and 14B total parameters. Its architecture follows a Mixture-of-Transformer-Experts design, with separate experts and visual encoders for understanding and generation, while sharing self-attention over the same token sequence. This design allows BAGEL to handle tasks such as visual question answering, text-to-image generation, image editing, and free-form visual manipulation in a single model.

InternVL-U. InternVL-U [30] is a lightweight 4B-parameter UMM for understanding, reasoning, image generation, and image editing. Instead of training a fully native UMM from scratch, it builds on a strong MLLM backbone and adds a specialized MMDiT-based visual generation head. The model follows a unified contextual modeling design with modality-specific modules and decoupled visual encoders. This design aims to preserve the semantic strength of the MLLM while enabling high-quality generation and editing. InternVL-U also uses reasoning data synthesis, including text-rich and scientific tasks, to better align user intent with fine-grained visual outputs.

C.2 Dataset and Benchmark Usage

We list the name, the number of samples, and the question types of utilized benchmarks in our main text in Table 9.

Table 9: Benchmarks utilized in our main text.

Name	Num Samples	Type
MME [8]	2,374	visual question answering
MMBench (dev) [19]	4,329	visual question answering
MMVP [31]	300	pair-level visual question answering
RealWorldQA [36]	765	real-world visual question answering
GEdit (EN) [18]	606	image editing
RISE-Bench [45]	360	reasoning, logical or spatial image editing
IntelligentBench [7]	350	reasoning, logical or spatial image editing

C.3 Hyper-parameter Choosing

For all models and baselines, we use the default settings and hyper-parameters described in their original repositories and papers. Notably, we keep the random seed unchanged (42) for all baselines. For G^{2}TR, we keep ρ to 0.5 as described in our paper. We then set $K_{\min}$ to 1.

For understanding tasks of BAGEL-7B without thinking and InternVL-U, we set the max response token length to 20. For BAGEL with thinking mode, we set the max response token length to 2048 to avoid truncation. For editing tasks, we set the time steps to 50, and other settings are default ones.

D Theoretical Analysis

Let $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^{N}$ denote the visual tokens from the understanding encoder, and let $\mathbf{Z} = E_{\text{vae}}(\mathbf{x})$ denote the latent grid produced by the generation-side VAE. In unified multimodal models, visual tokens need to support both semantic understanding and visual generation.

However, these two objectives may prefer different visual granularity. This motivates recent UMM designs that decouple visual pathways for understanding and generation [34, 30]. In this view, the VAE latent space provides a compact representation that is directly optimized for image reconstruction and generation.

We therefore use the generation-side latent as a guidance signal for understanding-side token reduction. For each ViT token $\mathbf{u}_i$, we find its corresponding VAE latent anchor $\mathbf{z}_{c(i)}$, where $c(i)$ denotes the

token-to-latent-anchor mapping. We define the latent-alignment score as

$$s_i = \cos\big(\mathbf{u}_i,\, \mathbf{z}_{c(i)}\big)\,. \tag{14}$$

This score is a simple and training-free surrogate for the shared visual factors between the understanding token and the generation latent. Tokens with larger s_i are more consistent with the generation-side representation, and are therefore more likely to preserve visual details that matter for image editing and generation.

Ideally, the retained subset S should preserve the generation-relevant information in $\mathbf{U}$:

$$S^\star = \arg\max_{|S|=K} I(\mathbf{U}_S; \mathbf{Z}). \tag{15}$$

Directly optimizing this objective is intractable during inference. We instead optimize a modular surrogate:

$$R(S) = \sum_{i\in S} s_i, \qquad |S| = K. \tag{16}$$

In practice, we further encourage generation-related information over the VAE latent grid, rather than selecting all tokens from a few highly scored regions. This prevents the retained tokens from collapsing to local salient areas and helps preserve the global image layout.

After token selection, the removed tokens are not simply discarded. Each removed token is assigned to its nearest retained token in feature space and merged into it. Let $\tilde{\mathbf{U}}_S$ be the compressed token set, and let $\mathcal{R}_S(\tilde{\mathbf{U}}_S)$ denote its expanded approximation in the original token space. The induced token reconstruction error is

$$\epsilon_{\text{merge}} = \left\| \mathbf{U} - \mathcal{R}_S(\tilde{\mathbf{U}}_S) \right\|_2 . \tag{17}$$

This term measures how well the compressed tokens preserve the original visual token set after merging.

We can then state a conditional perturbation bound. Suppose the following UMM decoder $F(\cdot)$ is locally L-Lipschitz with respect to its visual-token input. Then

$$d\Big(F(\mathbf{U}), F(\mathcal{R}_S(\tilde{\mathbf{U}}_S))\Big) \le L \left\| \mathbf{U} - \mathcal{R}_S(\tilde{\mathbf{U}}_S) \right\|_2 , \tag{18}$$

where $d(\cdot,\cdot)$ denotes the output discrepancy. This bound shows that the output change is controlled by the token reconstruction error. Moreover, the selection score $R(S)$ controls which information is preserved before merging: VAE-guided selection keeps tokens that are more aligned with the generation latent, while token merging reduces the feature distortion caused by removing the remaining tokens.

Therefore, the method provides a generation-aware form of token reduction. Unlike attention-only or text-similarity-based pruning, which mainly preserves tokens useful for language-side prediction, VAE-guided reduction uses the generation branch itself as the relevance signal. This makes the retained tokens more consistent with the visual factors required by image generation, while still reducing the number of understanding-side visual tokens before LLM prefill.

E More Visualization Results

We put more visualizations of image editing results in Figure 7. Images and instructions are from IntelligentBench [7] and KRIS-Bench [35]. Our method still keeps necessary visual details in the edited images, and follow the instructions well, better than compared baselines.

F Special Note

This paper was finished during a very difficult time. My mother was unwell and I cannot be with her. Thank my mother three thousand for bringing me up and teaching me many things.

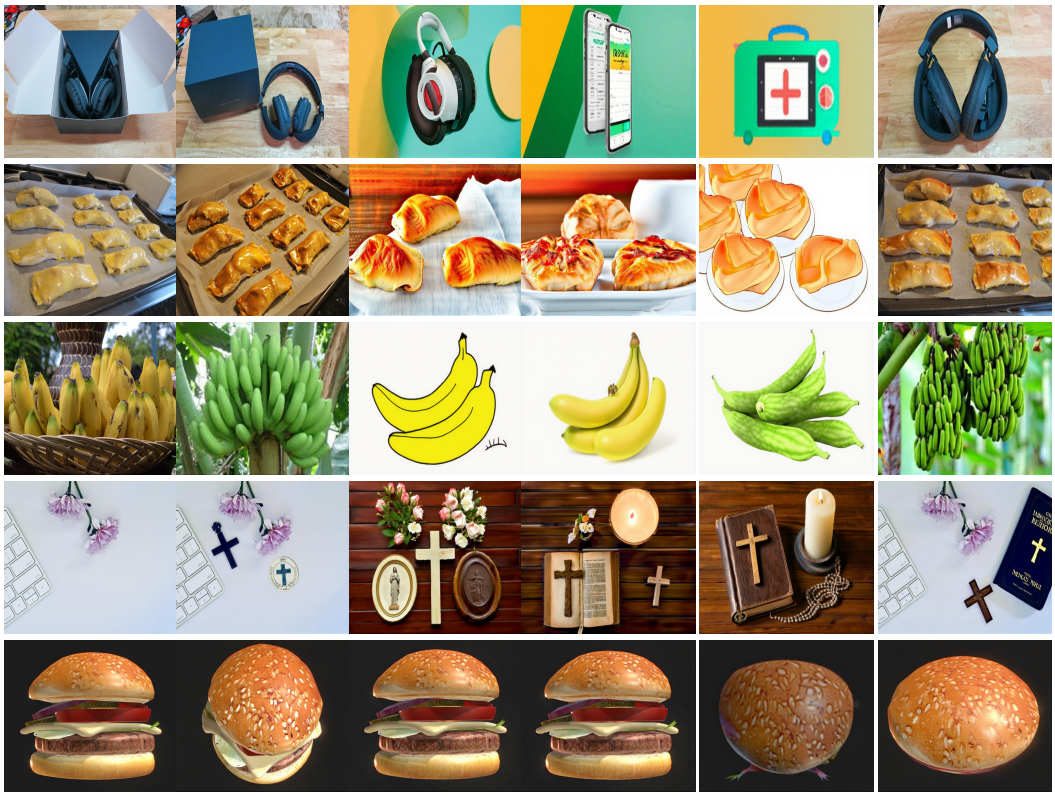

Figure 7: More visualization results. Instructions from top to down are: “What does the item in the box look like when taken out?”; “What is the appearance of these pastries after they have been baked?”; “The fruit in the image is ready for harvest. Could you display how it looks before it is ready?”; “Place Christian classic symbol items on the table.”; “Draw the top view of the object based on its front view.”. Our method still indicates strong performances.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly shows that we are doing understanding-side visual token reduction for UMMs. We propose a generation-guided method for that and conduct extensive experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a "Limitations and Future Work" Section in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We do not claim formal theoretical results that require theorem-level proofs. Appendix D provides a theoretical analysis of the proposed design and a conditional perturbation analysis under a local Lipschitz assumption.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We introduce the models, baselines, benchmarks we use, the GPU settings and so on in the main text and the Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The abstract has provided a link to our open-source codes. All experiments are done on open-source benchmarks, and we have disclosed all of them in Section 5.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We do not include training. For testing, we show the hyper-parameters in Appendix C. As described in the main paper, for baselines we use the default hyper-parameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We rely on UMMs, which are quite large, to conduct experiments. Consistent with prior works [45, 27, 7] and so on, we only report the scores on benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Since this is a work about efficient UMMs, we conduct experiments on efficiency in Table 3. This table includes metrics about computational cost.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We are sure that we obey the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We do not discover any negative social impacts of our work. We discuss the positive impact: benefiting UMM deployment under resource-constraint scenarios.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper does not involve such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the creators and original owners of the datasets, benchmarks, pretrained backbones, and baseline methods used in the experiments through corresponding citations. We use these existing assets in accordance with their intended research purposes and respect their stated licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release a new dataset, benchmark, model checkpoint, or standalone asset. The main contribution is the G^2TR method, whose details and experimental settings are documented in Section 4, Section 5.1, and Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.