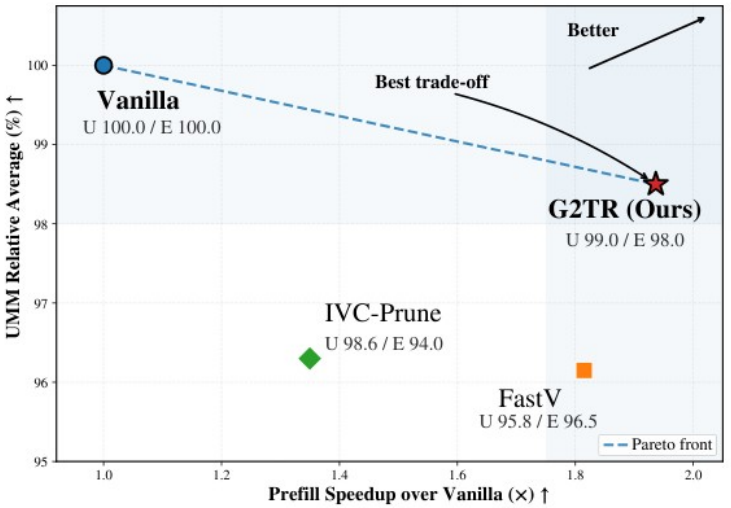
NEURIPS 2026 · RESEARCH PRESENTATION

G²TR

Generation-Guided Visual Token Reduction for Separate-Encoder Unified Multimodal Models

Junxian Li, Kai Liu, Zizhong Ding, Zhixin Wang,
Zhikai Chen, Renjing Pei, Yulun Zhang

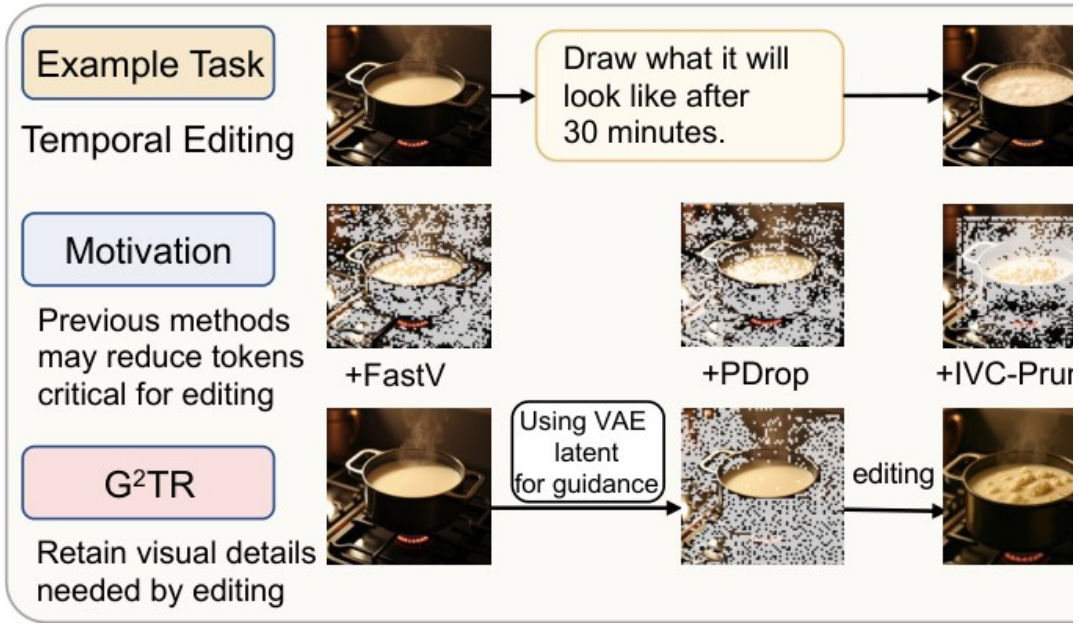
Shanghai Jiao Tong University · Huawei Technologies



50% visual tokens · 1.94× fewer prefill FLOPs

Visual token reduction in UMMs has a different objective

Existing pruning signals mainly target recognition and reasoning.



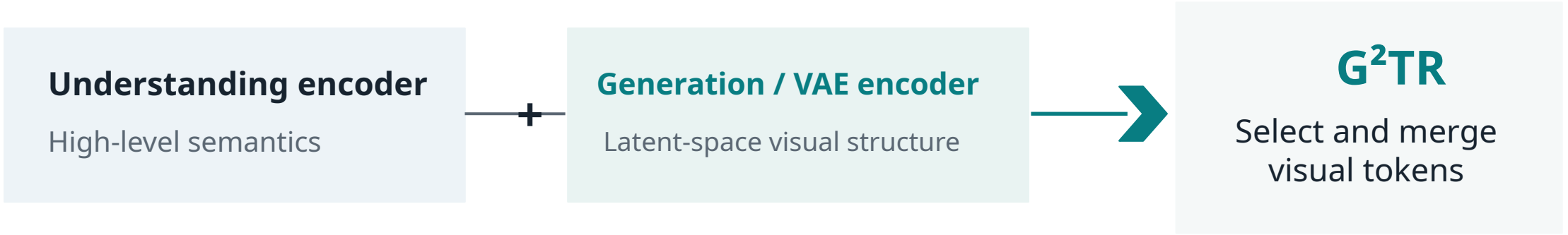
The missing requirement

A token that appears unimportant for answering a question may still carry details needed for image editing.

Understanding + Editing

Key insight: use the generation branch as guidance

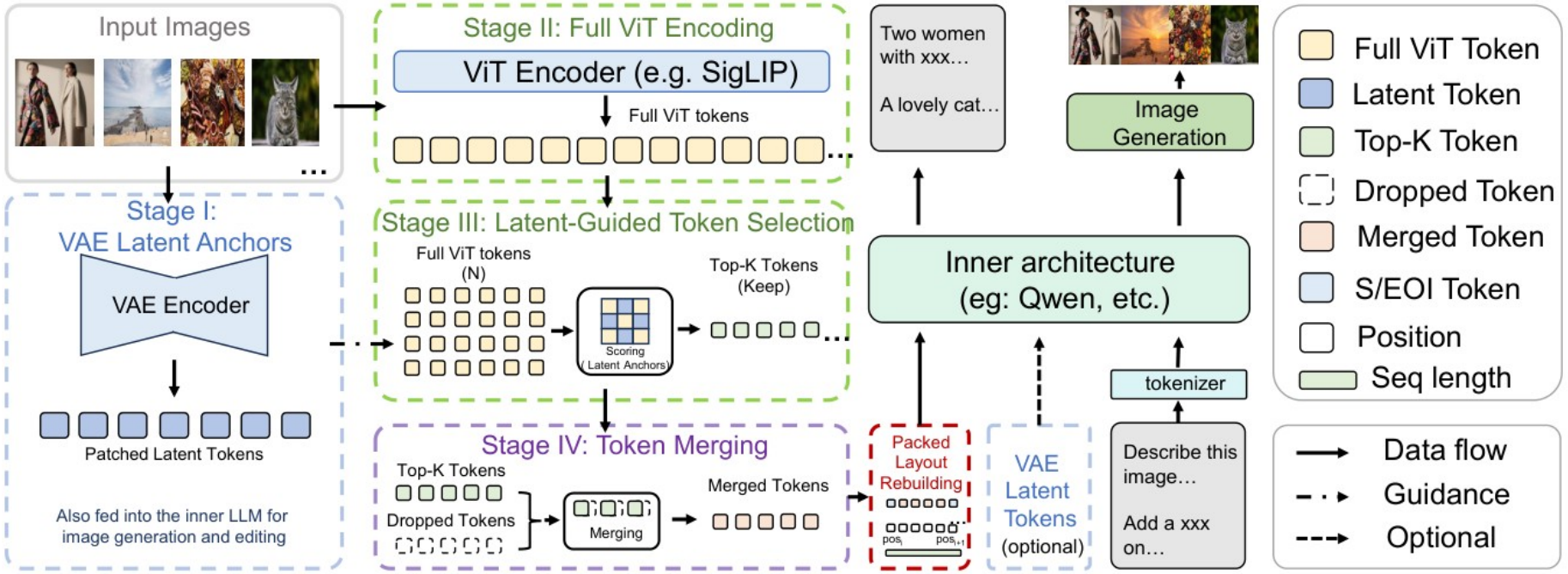
Separate-encoder UMMs already expose two complementary views of the image.



Rank understanding tokens by consistency with generation latents.

No additional training · no modification to inner LLM attention

G²TR: latent-guided selection with information recovery



I. VAE latent anchors | **II. Full ViT encoding** | **III. Balanced selection** | **IV. Token merging**

Only understanding-side tokens are compressed before LLM prefill.

Qualitative results: editing details survive compression

At 50% retained visual tokens, G²TR preserves fine-grained image changes.



Original comparison from the paper: Input · Vanilla · FastV · PDrop · IVC-Prune · G²TR

A stronger understanding–editing trade-off

BAGEL-7B · 50% visual token budget

1.94×

Prefill FLOPs reduction

4.1325T → 2.1341T

99.0%

Understanding retained

relative to vanilla

98.0%

Editing retained

relative to vanilla

Among compressed baselines, G²TR achieves the strongest relative averages on both understanding and editing in BAGEL-7B.

InternVL-U: 94.9% relative understanding average at a 50% token budget.

Takeaway: compression should respect generation

Generation-guided visual token reduction

Why	Understanding-only cues can miss editing-critical details.
How	VAE alignment → balanced selection → token merging.
Result	50% fewer visual tokens, 1.94× fewer prefill FLOPs, strong task quality.

Code: github.com/lijunxian111/G2TR