

IF-Prune: Information-Flow Guided Token Pruning for Efficient Vision-Language Models

Guohao Sun^{1,2,*}, Yufei Wang^{1†}, Sizhuo Ma¹, Yuege Xie¹,
Yuting Cheng¹, Zhiqiang Tao^{2†}, Jian Wang^{1†}
¹Snap Inc. ²Rochester Institute of Technology

Abstract

Vision-language models (VLMs) with dynamic-resolution vision encoders achieve strong performance but face significant efficiency challenges due to long input sequences. A common approach is to assess the importance of tokens and prune those that are less informative. Recent methods that use a small VLM to generate importance maps for visual tokens have outperformed existing rule-based and similarity-driven pruning approaches, particularly at high pruning ratios. However, directly using the small VLM remains unreliable, as it relies on aggregated visual attention weights as an importance score, which can lead to noisy guidance when the generated tokens are incorrect. To address this, we invert the approach by having it detect non-informative visual tokens based on the user’s query. By adding a variational information bottleneck to the small VLM, we can approximate the entropy of each visual token to provide pruning guidance. Such a posterior-guided pruning method enables the large VLM to retain its reasoning capacity while improving efficiency. Extensive experiments on eight benchmarks demonstrate the effectiveness of our approach. With only 5% of visual tokens retained, the large VLM preserves 95% of its original performance, outperforming the state of the art by 8%. The code is available at <https://github.com/snap-research/EVLM-IF-Prune>.

1. Introduction

Vision–language models (VLMs) [4, 10, 21, 29, 30] have demonstrated remarkable progress across a wide range of visual tasks, yet their deployment remains hindered by high computational costs. A key source of inefficiency arises from the large number of visual tokens produced by dynamic resolution encoders, which significantly increase

the sequence length and burden downstream reasoning [4]. However, the visual input has high redundancy and sparsity as a generation condition in VLM. Therefore, token pruning [5, 22, 24, 38, 39] has emerged as a promising strategy for improving efficiency by discarding less informative visual tokens.

Despite recent progress, existing pruning approaches suffer from fundamental limitations. For example, FastV [8] assumes that cross-attention from the first generated token provides a reliable signal of token importance. In practice, however, this assumption often breaks down, leading to unstable pruning decisions. More recent work, SGP [43], aggregates attention scores across all generated tokens from a small VLM to construct importance maps, which are then used to guide pruning of a larger VLM with the same architecture. While this strategy yields improvements at high pruning ratios, its pruning guidance is heavily dependent on the small model’s prior knowledge. This reliance limits generalization to complex instructions with higher visual dependencies. As shown in Fig. 1a, when the small-VLM lacks the prior knowledge to answer a given query, the importance map it produces becomes ineffective, resulting in noisy token retention and impairing the large-VLM’s reasoning capacity.

To address this, we invert the paradigm: rather than asking a small model with limited ability to identify the most important visual tokens and forcing the large model to follow, we instead train the small model to approximate the distribution of non-informative tokens. Inspired by variational information bottleneck [2, 32, 34], we train the small-VLM learns to map visual tokens conditioned on the user input to a latent space via a light-weight module, as shown in Fig. 1b. By restricting the information flow of visual inputs through a KL regularization term, the visual tokens with low entropy are treated as non-important ones, which will be hard-pruned before passing to the large-VLM at inference time. In this way, the small-VLM can highlight broader regions of informative content than the attention-based guidance, making the pruning guidance less deterministic but more inspirational and auxiliary [15], and en-

*Work done as intern at Snap Inc.

†Corresponding authors: Zhiqiang Tao (zhiqiang.tao@rit.edu), Yufei Wang (ywang25@snap.com, im.wangyufei@gmail.com), and Jian Wang (jwang4@snap.com)

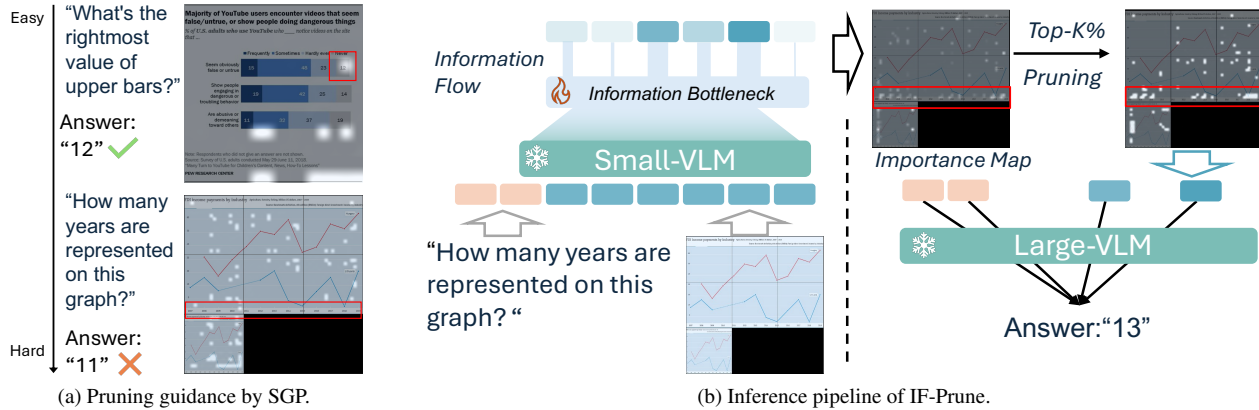


Figure 1. (a) SGP utilizes a pre-trained VLM for the importance map prediction, but failed to provide helpful pruning guidance due to its answer-driven mechanism. (b) We fine-tuned an information bottleneck module to map the output visual embeddings from a small-VLM to a latent variable, which are used to compute the importance of each visual token given the provided text prompt. The pruning guidance is more helpful than SGP after top- $K\%$ pruning for the large-VLM.

abling the large-VLM to retain its full reasoning capability with better efficiency.

Overall, we posit that effective pruning requires moving beyond answer-driven heuristics and instead estimating token importance through a principled probabilistic framework. This work introduces **IF-Prune**: Information-Flow Guided Token Pruning for Efficient Vision-Language Models, which formulates token importance estimation as an amortized variational inference problem in VLM [31]. Specifically, we fine-tune a small-VLM to act as a latent variable sampler [18], and approximate the distribution over each visual token’s contribution to the downstream task by parameterizing each latent visual token. Intuitively, this posterior-driven formulation yields pruning guidance that is both query- and answer-aware, ensuring that the retained visual tokens are more informative for reasoning. By computing the KL-divergence between the predicted posterior and the prior distribution as an importance score, IF-Prune leverages the small-VLM to provide a robust and transferable estimate of visual relevance.

IF-Prune is also practical for inference. Prior methods [3, 5, 8] compute importance inside the large VLM’s decoder or require full decoding to an end-of-sequence token, incurring significant overhead. In contrast, IF-Prune produces guidance in a single forward pass of the small VLM, substantially reducing the large model’s FLOPs and memory without sacrificing accuracy. Furthermore, our method works well with optimized libraries, such as FlashAttention [12], since it does not require explicitly outputting all the attention weights, unlike SGP [43] and FastV [8]. We summarize the contributions of this work as follows.

- We introduce IF-Prune, a principled probabilistic framework for information-flow-guided token pruning, which casts visual token importance estimation as an amortized variational inference problem, moving beyond answer-

driven attention heuristics.

- We propose a lightweight, one-pass adaptive pruning mechanism based on a fine-tuned small VLM that can be directly applied to various large models without additional training.
- Extensive experiments demonstrate that IF-Prune achieves state-of-the-art trade-offs between efficiency and performance. Specifically, IF-Prune retains up to 95% of the original accuracy while utilizing only 5% of the visual tokens, resulting in a 40% reduction in computational cost, and consistently outperforms previous SOTA methods by 7% across eight benchmarks.

2. Preliminaries

Vision-Language Models. General-purpose vision-language models (VLMs) are typically instantiated as causal large language models (LLMs) conditioned on visual inputs. To strengthen visual understanding, recent families such as LLaVA [21, 30], QwenVL [4, 35], and InternVL [9, 10] adopt dynamic visual encoders. A high-resolution image is tiled into multiple crops; each crop is passed through a ViT to produce a fixed-length sequence of patch embeddings. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote textual embeddings and $\mathbf{V} \in \mathbb{R}^{m \times d}$ denote visual embeddings, both projected into the LLM’s d -dimensional space, and let $N = m + n$. The LLM consumes the concatenated sequence $[\mathbf{V}; \mathbf{X}] \in \mathbb{R}^{N \times d}$ and outputs contextualized hidden states $[\mathbf{V}', \mathbf{X}']$, in which visual representations mutually inform text information. At inference time (pre-fill), the VLM forms a single causal sequence over visual and textual tokens. With an appropriate attention mask, each visual token can attend to previously seen text (e.g., user instructions and system prompts), allowing the visual features to be shaped by the query context.

Small-VLM-Guided Visual Token Pruning. While ex-

tending the visual-token sequence length can enhance fine-grained perception, a substantial portion of tokens is empirically redundant or unhelpful. Selecting only the most informative tokens, therefore, poses a fundamental efficiency–accuracy trade-off [3, 8, 42]. Small-VLM–Guided Pruning (SGP) [43] tackles this problem by generating a token-level *importance map* $\mathbf{s} \in \mathbb{R}^m$ using a compact vision-language model (VLM), assigning each visual token an importance score. Specifically, SGP derives \mathbf{s} by aggregating the attention weights between all generated tokens and the conditional visual tokens. In contrast, our proposed IF-Prune estimates \mathbf{s} in a single forward pass, enabling more direct and efficient importance inference. The top- $K\%$ most salient visual tokens are then retained, while the remaining tokens are hard-pruned before being forwarded to the LLM decoder of the answer predictor.

3. Posterior-Guided Visual Token Pruning via Variational Inference

3.1. Variational Information Bottleneck in VLMs

A central challenge in visual token pruning is determining the relative importance of each token without introducing significant computational overhead. We propose a *token-wise variational information bottleneck* framework that treats each visual token as a stochastic latent variable and uses Kullback–Leibler (KL) divergence to quantify its information contribution.

Formally, given a set of visual embeddings $\mathbf{V}' = \{\mathbf{V}'_1, \dots, \mathbf{V}'_m\}$ after the small-VLM forward pass, we map them into latent variables $\mathbf{Z} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_m\}$, where each $\mathbf{Z}_i \sim Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i)$ is the latent representation of the i^{th} visual token. Each token-wise latent variable is represented by a multivariate Gaussian distribution as:

$$Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i) = \mathcal{N}(\mu_\theta(\mathbf{V}'_i), \sigma_\theta^2(\mathbf{V}'_i)), \quad (1)$$

where $\mu_\theta(\mathbf{V}'_i), \sigma_\theta^2(\mathbf{V}'_i) \in \mathbb{R}^d$ are predicted by a projection layer parameterized by θ . Specifically, in sequence-to-sequence LLM with causal attention, \mathbf{V}' can naturally fuse the prior query information by cross-attention over \mathbf{X} as shown in Fig. 2. In this way, by conditioning on \mathbf{V}' , the latent variable prediction is implicitly conditioned on both query and visual information.

To encourage disentanglement across channels, we adopt a learnable prior distribution $P(\mathbf{z}) = \mathcal{N}(\mu_p, \sigma_p^2)$, where $\mu_p, \sigma_p^2 \in \mathbb{R}^d$ are per-channel learnable mean and variance that are shared over the whole training data space. This design allows certain latent dimensions to carry more informative content while encouraging redundancy reduction in less important dimensions.

To this end, the KL divergence between the approximate

posterior and prior,

$$\begin{aligned} D_{\text{KL}}(Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i) \| P(\mathbf{z})) \\ = \frac{1}{d} \sum_{j=1}^d D_{\text{KL}}(Q_\theta(\mathbf{Z}_i^{(j)} | \mathbf{V}'_i^{(j)}) \| P(\mathbf{z}^{(j)})), \end{aligned} \quad (2)$$

representing the average amount of channel-wise information the i^{th} visual token contributes beyond the prior belief with d dimension. Intuitively, tokens that deviate strongly from the prior carry higher task-relevant information, while tokens with near-prior distributions contribute little. Thus, the KL divergence naturally serves as a token-wise *importance score*, forming the basis for pruning guidance at inference time.

Instead of directly predicting the posterior mean of the i^{th} visual token (i.e., $\mu_\theta(\mathbf{V}'_i)$) using a projection layer, we introduce a channel-wise gating mechanism to enhance the expressivity of the posterior mean as:

$$\mu_\theta(\mathbf{V}'_i) = \sigma(I_\theta(\mathbf{V}'_i)) \odot (\mathbf{V}'_i - \mu_p) + \mu_p, \quad (3)$$

where $I_\theta(\mathbf{V}'_i)$ is a learned channel-wise importance gate, $\sigma(\ast)$ is a sigmoid function, and \odot denotes element-wise multiplication. This mechanism enables the model to independently modulate how much information each channel contributes to the posterior within a bound, since $0 < \sigma(\ast) < 1$. So the gate upper-bounds how far the posterior mean can wander from the prior, directly capping the KL explosion and stabilizing optimization. However, a free mean projection can push $\mu_\theta(\mathbf{V}'_i)$ arbitrarily far, making the KL term volatile.

3.2. Reconstruction Objective

Our training objective extends the classical variational information bottleneck [2, 33] to the token level, which can be easily applied to token prediction within LLM. Overall, the goal of our objective function is twofold: **1** to ensure the accurate reconstruction of the target output conditioned on both the query and the latent visual tokens, and **2** to penalize redundant tokens by compressing their representations towards the prior.

Representing each visual token as a continuous Gaussian distribution makes the optimization gradient intractable. Following the **reparameterization trick** [20], we “reparameterize” the distribution $Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i)$ as,

$$\mathbf{Z}_i \sim Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i) = \mathcal{N}(\mu_\theta(\mathbf{V}'_i), \sigma_\theta^2(\mathbf{V}'_i)) \quad (4)$$

$$\mathbf{Z}_i = \mu_\theta(\mathbf{V}'_i) + \sigma_\theta(\mathbf{V}'_i) \cdot \epsilon, \quad (5)$$

where $\epsilon \in \mathcal{N}(0, I)$ is an auxiliary noise variable. Conditioning on the original query \mathbf{X} and the reparameterized latent visual tokens $\mathbf{Z} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_m\}$, the reconstruction loss aims to maximize the expectation of the log-likelihood

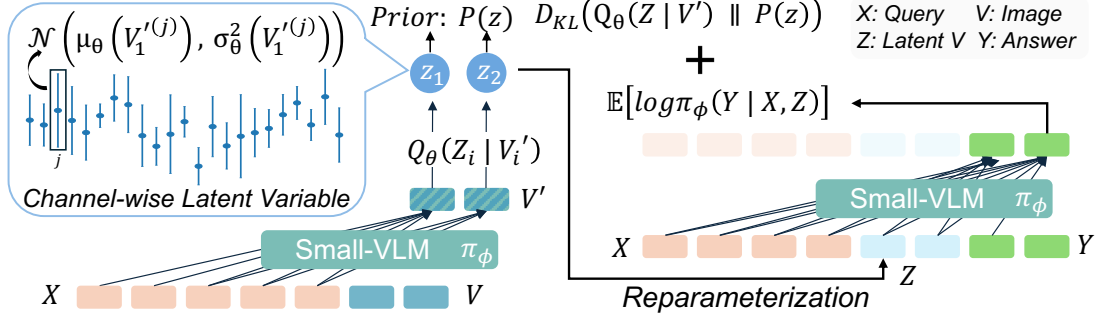


Figure 2. The overview of our training pipeline. For each data sample $\mathbf{X}, \mathbf{V}, \mathbf{Y} \sim \mathcal{D}$, we call small-VLM (i.e., π_ϕ) twice. 1^{st} **forward**: Input \mathbf{X} and \mathbf{V} , and the output \mathbf{V}' are mapped to latent space using $Q_\theta(\cdot)$, computing KL divergence with a shared prior. 2^{nd} **forward**: compute the cross-entropy between the predicted answer $\pi_\phi(\mathbf{Y} | \mathbf{X}, \mathbf{Z})$ and the ground truth \mathbf{Y} .

of the final answer \mathbf{Y} . To this end, the overall loss is:

$$\mathcal{L} = \underbrace{\mathbb{E}_{\mathbf{X}, \mathbf{Y} \sim \mathcal{D}, \mathbf{Z}} [\log \pi_\phi(\mathbf{Y} | \mathbf{X}, \mathbf{Z})]}_{\text{Reconstruction loss}} - \frac{\beta}{m} \sum_{i=1}^m \underbrace{D_{KL}(Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i) \parallel P(\mathbf{z}))}_{\text{Token-wise KL penalty}}, \quad (6)$$

where π_ϕ is the conditional likelihood modeled by the VLM with parameters ϕ , and β is a trade-off hyperparameter.

The reconstruction loss ensures the latent tokens preserve sufficient information for accurate answer prediction, while the KL term regularizes each token against the prior. By applying this penalty at the token level rather than the sequence level, we achieve two key benefits: 1. **Granular importance estimation**. Each token’s KL divergence reflects its marginal utility for the downstream task, enabling fine-grained pruning decisions. 2. **Adaptive compression**. The learnable per-channel prior allows the model to automatically retain highly informative latent dimensions while suppressing redundancy.

In summary, Eq. 6 enforces a principled token-wise trade-off between predictive sufficiency and compression. The resulting KL-based importance scores can be directly employed as a pruning criterion, yielding both interpretability and computational efficiency.

3.3. Posterior-Guided Visual Token Pruning

At inference time, we first pass the concatenated sequence of text and visual tokens (i.e., \mathbf{XV}) to a small-VLM (S-VLM). Then, we extract \mathbf{V}' from the last hidden-states of the output sequence ($\mathbf{X}'\mathbf{V}'$), where \mathbf{V}' are new visual embeddings containing query information. Next, we approximate the latent variable of each visual token (i.e., $\mathbf{Z}_i \sim Q_\theta(\mathbf{Z}_i | \mathbf{V}'_i)$), then compute the importance score for each visual token (i.e., $s \in \mathbb{R}^m$), which are used to guide the token pruning in large-VLM (i.e., L-VLM). This work computes importance map $s = \{D_{KL}(Q_\theta(\mathbf{Z}_1 |$

$\mathbf{V}'_1) \parallel P(\mathbf{z}), \dots, D_{KL}(Q_\theta(\mathbf{Z}_m | \mathbf{V}'_m) \parallel P(\mathbf{z}))\}$, where the posterior prediction layers (parameterized by θ) and the priors $P(\mathbf{z})$ were optimized in the training process. To be noticed, S-VLM should share the same model architecture as the L-VLM for consistent visual encoding. Following SGP [43], we retain the top- $K\%$ of visual tokens by the ranked vs . Then, we perform hard pruning on \mathbf{V} and the pre-computed position embeddings. In this way, the remaining visual tokens retain the original spatial information from the entire visual input.

4. Experimental Results

4.1. Training Recipe

This work uses the InternVL family of models for experiments, as their architecture is designed to take the full visual sequence without compression, thereby clearly demonstrating the effectiveness of our method. For the small-VLM (i.e., $\pi_\phi(\cdot)$), we initialize and fix the model parameters from the pretrained InternVL2.5-1B [10], followed by a learnable light-weight projection module (i.e., $Q_\theta(\cdot)$), which consists of two MLP layers, and two learnable embeddings (prior mean μ_p and variance σ_p^2). To alleviate the domain shift between $\pi_\phi(\mathbf{Y} | \mathbf{X}, \mathbf{V})$ and $\pi_\phi(\mathbf{Y} | \mathbf{X}, \mathbf{Z})$ during training, we fine-tune the small-VLM using Eq. 6 with LoRA for one epoch. Please refer to the supplementary for our detailed hyperparameters. For better generalizability, we follow the training data used in InternVL, which is a mixture of single-image instruction data proposed by ShareGPT-4V [6], LLaVA[21], and DVQA [19], etc.

4.2. Benchmarks and Baseline Pruning Methods

We consider the pruning guidance as a general-purpose assistant, and this work focuses on single-image tasks. For OCR and chart understanding, we utilize TextVQA [28] and ChartQA [25]. To validate the capabilities in real-world scenarios with open-form instructions, we utilize MMStar [7] and RealWorldQA. Besides general visual understanding,

visual perception evaluates the model’s reasoning ability, so we adopt MME [14], MMBench [23], MM-Vet [40], and GQA [16]. For a fair comparison with other pruning methods, we primarily report the baseline results of our own implementation and utilize lmms-eval [41] for consistency in the test setting. All our results are reported with greedy sampling and zero-shot prediction. We carefully choose four pruning methods to compare with ours, where ToME [5] solely focuses on reducing visual redundancy in the vision encoder, FastV [8] progressively reduces the number of visual tokens in the LLM forward process based on attention weights, and SGP [43] utilizes a small-VLM to provide pruning guidance, which is similar to our approach.

4.3. Comparing IF-Prune with Previous Methods

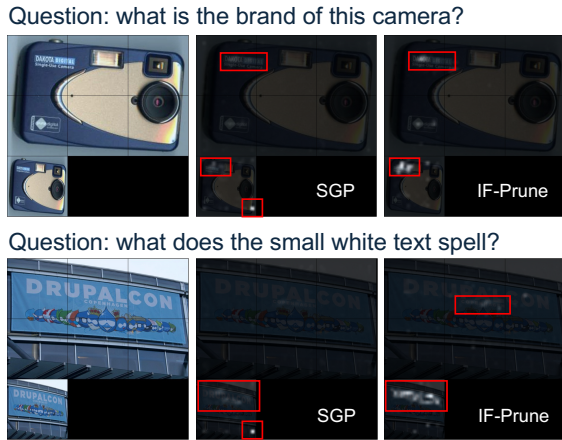


Figure 3. Visualization of visual token importance map proposed by SGP and IF-Prune (ours).

IF-Prune works as a soft visual cue detector with reasoning capability, rather than providing only direct, point-wise pruning guidance. Existing approaches, such as FastV and SGP, adopt answer-driven pruning strategies, in which the retained visual tokens are either directly tied to the predicted answer or largely irrelevant noise. This strong reliance on a VLM’s prior knowledge inherently limits both generalization and robustness. In contrast, by assigning higher importance scores to a broader set of potentially informative visual tokens (see Fig. 3), IF-Prune preserves critical contextual cues for downstream reasoning. This enables the large VLM to perform more precise and fine-grained inference, ensuring both reliability and trustworthiness in pruning decisions. As illustrated in Fig. 5, IF-Prune consistently identifies semantically meaningful and question-related visual cues, whereas SGP primarily localizes tokens directly tied to the predicted answer. We quantify the difficulty of human instructions by their degree of visual dependency, with more complex queries requiring a larger set of relevant visual tokens for accurate reasoning. Under this character-

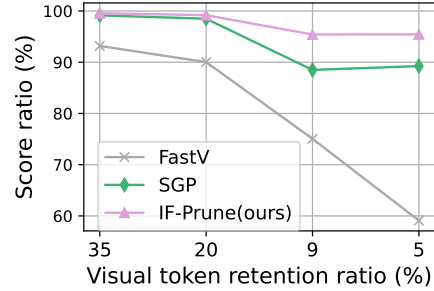


Figure 4. Performance–efficiency curve. IF-Prune demonstrates greater stability under progressively higher token pruning ratios, preserving accuracy more effectively.

ization, we observe that pruning decisions guided by SGP enable the large VLM to handle relatively simple queries but result in substantial performance degradation on visually demanding ones.

Quantitative results in Table 1 further validate this advantage. With only 20% of visual tokens retained, IF-Prune and SGP preserve 99.4% and 98.82% of the original performance (measured by the score ratio), respectively, indicating less than a 1% drop. However, under more aggressive pruning, the performance gap widens significantly: with just 5% of tokens, IF-Prune still maintains 95.4% of the full performance, while SGP and FastV degrade to 88.9% and 67.1%, respectively. Fig. 4 also shows that pruning guided by IF-Prune yields more stable performance than competing methods. Along with the qualitative examples in Fig. 5, we have presented our main hypothesis: preserving highly informative tokens is more effective than relying solely on answer-related tokens. More experimental results on LLaVA-1.5 can be found in the supplementary.

4.4. One Can Serve Many

While using a small VLM for pruning guidance is effective, the need to fine-tune a separate small VLM for each large model is practically inefficient. To assess the generalizability of our approach, we examine whether a single fine-tuned small VLM can transfer pruning guidance to larger models with the same architecture. Reusing the guide model from Table 1 (InternVL2.5-1B), we prune InternVL2-8B with $L = 0$ and report results in Table 2. At a moderate retention rate ($K = 20\%$), both SGP and IF-Prune remain close to the full-token baseline (normalized scores of 98.29% and 97.56%, respectively). Under aggressive pruning ($K = 5\%$), IF-Prune clearly surpasses SGP at the same retention: 94.03% vs. 90.34% (+3.69). Gains are largest on MMBench and MMStar. These findings are consistent with Table 1 and indicate that the amortized posterior learned by the small VLM transfers reliably across model scales within the InternVL family. In practice, we highlight the $K = 5\%$ setting for its superior performance–efficiency trade-off.

Table 1. Comparison of InternVL2-26B with different visual token pruning methods. After obtaining the importance map using different methods, including FastV, SGP, and IF-Prune, we retain the top- $K\%$ (i.e., token ratio) of all input visual tokens and execute hard pruning at the L^{th} decoder layer of InternVL2-26B. † are results based on our reproduced experiments. The best results are **bold**.

Method	K	L	TextVQA	ChartQA	GQA	MMStar	MMBench	MM-Vet	MME	RealWorldQA	Score ratio ↑
			val	All	test-dev	test	en-dev	test	test	test	
InternVL2-26B	100%	-	82.45	84.92	64.89	60.08	83.46	64.00	2270	67.58	100.00%
ToMe	20%	9	75.74	62.44	63.61	-	81.82	52.50	2178	-	94.88%
FastV†	20%	9	75.62	71.68	61.20	53.01	78.31	45.00	2140	63.27	93.18%
SGP†	20%	9	81.97	81.68	64.62	56.77	80.76	62.34	2258	67.50	99.15%
IF-Prune (ours)	20%	9	81.48	82.60	64.56	57.46	80.58	61.01	2271	66.14	99.55%
FastV†	20%	0	73.42	67.32	60.68	50.55	78.26	52.66	2110	60.26	90.03%
SGP†	20%	0	81.14	80.92	64.70	56.97	80.50	61.33	2252	67.90	98.49%
IF-Prune (ours)	20%	0	81.28	82.36	64.86	56.45	79.98	60.32	2263	66.54	99.19%
ToMe	5%	2	51.69	28.60	57.52	-	73.09	37.70	1933	-	82.33%
FastV†	5%	2	43.84	26.10	44.90	32.65	62.33	31.60	1799	44.05	75.05%
SGP†	5%	2	78.70	71.08	62.04	50.92	73.71	49.82	2007	64.84	88.50%
IF-Prune (ours)	5%	2	79.24	71.12	63.52	53.10	77.58	50.83	2189	65.62	95.41%
FastV†	5%	0	20.06	24.64	43.41	32.65	36.94	21.74	1418	44.05	59.10%
SGP†	5%	0	78.77	70.68	62.08	50.62	73.28	50.23	2028	65.10	89.25%
IF-Prune (ours)	5%	0	79.04	70.96	63.53	52.49	77.23	51.42	2190	66.01	95.44%

Table 2. Performance comparison of InternVL2-8B with different pruning methods including SGP and IF-Prune.

Method	K	GQA	MMStar	MMBench	RealWorldQA	Score %
InternVL2-8B	100%	62.70	59.11	81.90	65.10	100%
SGP	20%	62.59	56.37	80.67	64.58	98.29%
IF-Prune	20%	62.54	56.93	79.64	63.14	97.56%
SGP	5%	59.95	50.37	71.22	61.31	90.34%
IF-Prune	5%	58.47	53.34	76.46	62.48	94.03%

These results indicate that a single fine-tuned small model can effectively guide pruning across diverse large VLMs, offering clear advantages over attention-based methods such as SGP and FastV. Rather than requiring model-specific attention-implementation for each architecture, IF-Prune provides a reusable, semantically meaningful pruning signal that transfers reliably across models of different scales. Consequently, it enables efficient, interpretable, and scalable compression of large VLMs without repeated fine-tuning or exhaustive full-model analysis.

4.5. Efficiency-Performance Trade-off

Although IF-Prune employs a small VLM to guide token pruning for the large VLM, the overall computational cost of the system (small + large models) remains lower than running the large VLM alone. Table 3 reports the average FLOPs measured over 100 samples. Compared with SGP, IF-Prune requires significantly fewer FLOPs in the small model because it only performs a single forward pass to estimate token importance, whereas SGP must generate a full response sequence.

Table 3. Performance and FLOPs of different pruning methods. We prune $100 - K(\%)$ of visual tokens at L^{th} decoder layer of L-VLM. S-F and L-F indicate the inference FLOPs of the small- and large-VLM. IF-Prune reduces overall FLOPs (small and large models) by 40% while maintaining 95% performance.

Method	K	L	S-F	L-F	FLOPs % ↓	Score % ↑
InternVL2-26B	100%	-	-	117.7T	100.0%	100%
SGP	20%	9		81.4T	81.5%	99.15%
	5%	2	14.5T	67.5T	69.7%	88.50%
	5%	0		65.4T	67.9%	89.25%
IF-Prune	20%	9		83.4T	74.6%	99.55%
	5%	2	4.7T	69.3T	62.9%	95.41%
	5%	0		67.3T	61.2%	95.44%

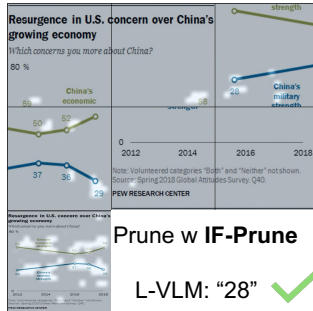
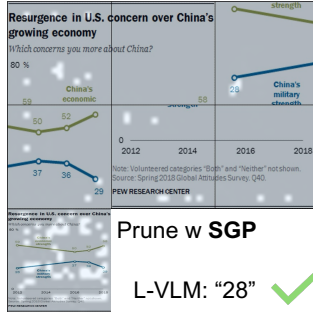
Table 4. Latency analysis of SGP and IF-Prune over 5k samples.

Method	K	Prefill (ms) ↓	Decode (ms) ↓	Throughput (token/s) ↑
InternVL2-8B	100%	229.1	55.8	16.9
SGP	5%	524.5	51.3	16.4
IF-Prune	5%	238.5	47.6	19.5

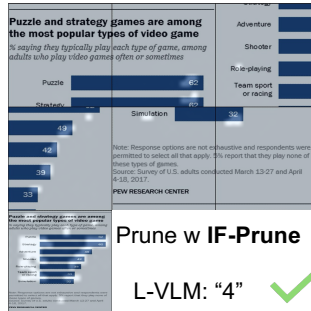
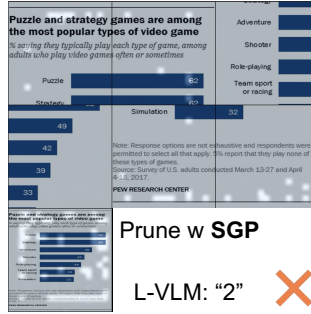
While IF-Prune introduces slightly higher FLOPs in the large VLM due to longer generated responses, the overall system cost remains substantially lower. For example, when pruning to $K = 5\%$ tokens at layer $L = 2$, IF-Prune reduces the total FLOPs to 62.9% of the baseline while preserving 95.41% of the performance. Across different pruning configurations, IF-Prune consistently achieves a better efficiency–performance balance than SGP.

While both methods reduce overall FLOPs during inference, FLOPs do not always directly translate to real-world efficiency. Therefore, we further analyze the latency of the prefill and decoding stages. As shown in Table 4, the prefill

1. "What's the least value of blue graph?"



2. "How many games in the chart have over 40 ratings?"



3. "Is the average of two extreme values greater than the middle bar value?"

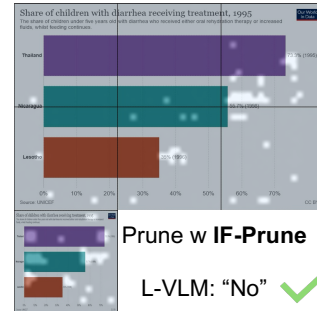
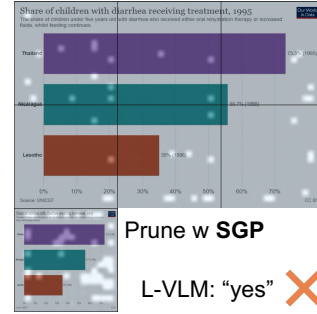


Figure 5. Comparison of the same large-VLM (L-VLM) with different pruning methods. For each visual input, we highlight the top-5% of all the visual tokens based on the importance map predicted by SGP and IF-Prune. **Upper three:** SGP provides answer-driven pruning guidance, impacting the large-VLM’s performance. **Lower three:** IF-Prune provides posterior-driven guidance, where the retained visual tokens are high query and answer relevance, allowing the L-VLM to perform sufficient visual understanding before answering.

stage accounts for most of the total inference time because it processes the entire input sequence. IF-Prune invokes the small VLM only once to estimate pruning guidance, resulting in a prefill latency close to that of the baseline. In contrast, SGP requires autoregressive generation from the small model, leading to significantly higher prefill overhead. After pruning the input of large-VLM guided by the small-VLM, both methods reduce decoding latency by processing fewer visual tokens. However, IF-Prune achieves lower overall latency and higher throughput.

4.6. Ablation Study

Effect of channel-wise gating activation. We compare the exponential and sigmoid (σ) functions as channel-wise gating activations to restrict the information flow in non-informative tokens. While both methods achieve non-trivial compression, sigmoid consistently outperforms exponential gating across benchmarks (90.80% vs. 89.83% overall score). This suggests that sigmoid provides a more independent and stable allocation of importance weights across channels, as its normalized scaling is from (0, 1). This reduces the gradient spikes and variance in the KL term, preventing the over-amplification of individual channels.

Training with adaptive KL weight. Our objective in Eq. 6 has a KL penalty term, aiming to compress the in-

formation of non-query and non-answer correlated visual tokens. A fixed coefficient β can, however, lead to suboptimal trade-offs across different training stages [13]. In the early phase of optimization, strong KL regularization may prematurely suppress informative tokens, hindering reconstruction fidelity. Conversely, weak regularization in later stages can lead to redundant token retention and slow convergence. To mitigate this issue, we adopt an *adaptive KL weighting* strategy. Concretely, we introduce a schedule $\beta(s) = \tau_{max} - (\tau_{max} - \tau_{min}) * \min(1, s/\gamma)$ is the annealing coefficient, s is the index of the current training step and γ is the number of warm-up steps. This scheme imposes fewer penalties at the beginning and stronger penalties later. As shown in Table 5, using a fixed $\beta = 0.5$ yields competitive performance, but a learnable prior with a linear schedule $\tau(0.2, 0.5)$ improves results by better balancing compression and reconstruction, indicating that adaptive KL weighting not only stabilizes training but also enables more precise pruning of redundant tokens.

5. Related Work

5.1. Information bottleneck

The Information Bottleneck (IB) [33] formalizes representation learning as a trade-off between task sufficiency

Table 5. Ablation study of major components in our proposed objective function for training the information bottleneck. We report the results of InternVL2-26B after pruning guided by our small-VLM using IF-Prune with $K = 5\%$ and $L = 0$ for all the models. β is the weight of the KL penalty and $f(*)$ is the gateway activation for posterior mean approximation.

β	$f(*)$	ChartQA	GQA	MMStar	MMBench	TextVQA	MM-Vet	RealWorldQA	Score % \uparrow
-	-	84.92	64.89	60.08	83.46	82.45	64.00	67.58	100%
0.5	exp	69.92	63.34	52.25	77.15	78.28	49.63	65.23	89.83%
0.5	σ	70.28	63.77	52.58	75.77	78.72	53.30	66.27	90.80%
$\tau(0, 1)$	σ	71.56	63.58	52.61	75.00	78.55	50.50	65.10	90.05%
$\tau(0.2, 0.5)$	σ	70.96	65.53	52.49	77.23	79.04	51.42	66.01	91.19%

and input compression [11]; its variational form (VIB) [2] makes this trainable at scale by regularizing a parametric posterior toward a simple prior through a KL term while maximizing predictive likelihood. IB/VIB has been widely used for compression and pruning by limiting per-unit information capacity, yielding task-aware sparsity across neurons, channels, and tokens, and often outperforming heuristic saliency measures at aggressive budgets. In Transformers, IB-style regularization has been applied to heads, MLP channels, and layers Wang and Yang [36], as well as to token representations in our study, where per-token latent variables are scored via KL-to-prior as a principled importance signal. Complementary “latent bottleneck” modules (e.g., resamplers or token learners) compress vision features into a compact, task-adaptive set [1], embodying the same retain-relevant/discard-nuisance principle even when not derived from the IB Lagrangian. Our approach, IF-Prune, instantiates an amortized, token-level IB for VLMs: a small VLM learns $Q_\phi(z|v)$ per visual token, and the per-token KL serves as the importance score.

5.2. Vision Language model

Vision-Language Models (VLMs) have advanced rapidly by aligning visual encoders with large language models, enabling multimodal reasoning across tasks such as image captioning, visual question answering, and video understanding. Early approaches such as CLIP [26] and ALIGN [17] demonstrated the power of contrastive pre-training, while more recent instruction-tuned models, including LLaVA [21, 30], QwenVL [4, 35], and InternVL [9, 10], leverage lightweight adapters or cross-attention modules to bridge modalities efficiently. These architectures typically concatenate visual tokens from a vision transformer with textual embeddings, enabling joint reasoning but also introducing substantial computational burdens when processing high-resolution images or long videos. The scaling of VLMs toward long-context multimodal understanding has further exacerbated these challenges, as visual tokens can dominate the sequence—often exceeding 80% of total tokens. Therefore, reducing the number of visual tokens without compromising semantic fidelity has emerged as a key research direction, motivating recent advances in token pruning and compression.

5.3. Visual Token Pruning and Compression

To address the inefficiency of processing redundant visual tokens, a wide range of token compression strategies has been proposed, spanning transformation-based, similarity-based, attention-based, and query-guided methods. Among these, attention-based pruning has attracted particular interest because it directly exploits sparsity in the attention maps of vision transformers or LLMs. Encoder-side methods, such as PruMerge+ [27] and VisionZip [38], select tokens with high attention relative to the [CLS] token and merge or discard the remainder. Decoder-side methods, including FastV [8] and PyramidDrop [37], prune inattentive tokens progressively across layers, guided by the average attention they receive from visual tokens. A line of work employs small-VLM to guide pruning. For instance, SGP [43] uses a small model to estimate token relevance by aggregating attention weights across all decoder layers during decoding, reducing the computational overhead of large backbones. While effective, these approaches face practical hurdles when integrated with optimized libraries such as FlashAttention, which obscure explicit attention scores.

6. Conclusions

This work introduces IF-Prune, a method that reframes visual token pruning in VLMs as an amortized variational inference problem rather than as attention- or answer-driven heuristics. By fine-tuning a light-weight information bottleneck, IF-Prune estimates token importance via the KL divergence between the approximated posterior and a prior distribution. This posterior-driven signal is both query- and answer-aware, enabling a small VLM to detect semantically relevant visual for the large VLM’s reasoning. Practically, IF-Prune produces pruning guidance in a single forward pass of the small-VLM, requires no architectural modifications to the large-VLM, and remains fully compatible with optimized kernels such as FlashAttention. Extensive experiments show that IF-Prune achieves state-of-the-art efficiency–accuracy trade-offs across diverse visual tasks. We believe IF-Prune offers a scalable, easily deployable approach to bring large VLMs closer to low-latency deployment without sacrificing reasoning quality.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and valuable suggestions. This work was supported by Snap Inc.

References

- [1] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 8
- [2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2016. 1, 3, 8
- [3] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 2, 3
- [4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 8
- [5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 5
- [6] 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*, 2024. 4
- [7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 4
- [8] 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 *European Conference on Computer Vision*, 2024. 1, 2, 3, 5, 8
- [9] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 2, 8
- [10] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024. 1, 2, 4, 8
- [11] Bin Dai, Chen Zhu, Baining Guo, and David Wipf. Compressing neural networks using the variational information bottleneck. In *International conference on machine learning*, 2018. 8
- [12] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 2
- [13] Miroslav Fil, Munib Mesinovic, Matthew Morris, and Jonas Wildberger. beta-vae reproducibility: Challenges and extensions. *arXiv preprint arXiv:2112.14278*, 2021. 7
- [14] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *ArXiv*, abs/2306.13394, 2023. 5
- [15] Jung-Ho Hong, Ho-Joong Kim, Kyu-Sung Jeon, and Seong-Whan Lee. Comprehensive information bottleneck for unveiling universal attribution to interpret vision transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1
- [16] 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*, 2019. 5
- [17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 2021. 8
- [18] Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. Inserting information bottlenecks for attribution in transformers. *arXiv preprint arXiv:2012.13838*, 2020. 2
- [19] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 4
- [20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3
- [21] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *ArXiv*, 2024. 1, 2, 4, 8
- [22] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 2024. 1
- [23] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, 2024. 5
- [24] Feipeng Ma, Hongwei Xue, Yizhou Zhou, Guangting Wang, Fengyun Rao, Shilin Yan, Yueyi Zhang, Siying Wu, Mike Zheng Shou, and Xiaoyan Sun. Visual perception by large language model’s weights. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1
- [25] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 4

- [26] 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*, 2021. 8
- [27] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 8
- [28] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 4
- [29] Guohao Sun, Can Qin, Huazhu Fu, Linwei Wang, and Zhiqiang Tao. Self-training large language and vision assistant for medical question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. 1
- [30] Guohao Sun, Can Qin, Jiamian Wang, Zeyuan Chen, Ran Xu, and Zhiqiang Tao. Sq-llava: Self-questioning for large vision-language assistant. In *European Conference on Computer Vision*, pages 156–172. Springer, 2024. 1, 2, 8
- [31] Guohao Sun, Hang Hua, Jian Wang, Jiebo Luo, Sohail A. Dianat, Majid Rabbani, Raghuvveer Rao, and Zhiqiang Tao. Latent chain-of-thought for visual reasoning. In *NeurIPS*, 2025. 2
- [32] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE information theory workshop (itw)*, 2015. 1
- [33] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 3, 7
- [34] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 2020. 1
- [35] 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. 2, 8
- [36] Yancheng Wang and Yingzhen Yang. Efficient visual transformer by learnable token merging. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 8
- [37] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *CVPR*, 2025. 8
- [38] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1, 8
- [39] Xubing Ye, Yukang Gan, Xiaohe Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1
- [40] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. 5
- [41] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024. 5
- [42] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. *arXiv preprint arXiv:2412.01818*, 2024. 3
- [43] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Zhikai Li, Yibing Song, Kai Wang, Zhangyang Wang, and Yang You. A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 3, 4, 5, 8