

A Stitch in Time Saves Nine: Small VLM is a Precise Guidance for Accelerating Large VLMs

Wangbo Zhao^{1*} Yizeng Han^{2*} Jiasheng Tang^{2,3} Zhikai Li¹ Yibing Song^{2,3}
 Kai Wang^{1†} Zhangyang Wang⁴ Yang You^{1†}

¹National University of Singapore ²DAMO Academy, Alibaba Group
³Hupan Lab ⁴The University of Texas at Austin

Abstract

*Vision-language models (VLMs) have shown remarkable success across various multi-modal tasks, yet large VLMs encounter significant efficiency challenges due to processing numerous visual tokens. A promising approach to accelerating large VLM inference is using partial information, such as attention maps from specific layers, to assess token importance and prune less essential tokens. However, our study reveals three key insights: (i) Partial attention information is insufficient for accurately identifying critical visual tokens, resulting in suboptimal performance, especially at low token retention ratios; (ii) Global attention information, such as the attention map aggregated across all layers, more effectively preserves essential tokens and maintains comparable performance under aggressive pruning. However, the attention maps from all layers require a full inference pass, which increases computational load and is therefore impractical in existing methods; and (iii) The global attention map aggregated from a small VLM closely resembles that of a large VLM, suggesting an efficient alternative. Based on these findings, we introduce a **training-free** method, **Small VLM Guidance for accelerating Large VLMs (SGL)**. Specifically, we employ the attention map aggregated from a small VLM to guide visual token pruning in a large VLM. Additionally, an early exiting mechanism is developed to fully use the small VLM’s predictions, dynamically invoking the larger VLM only when necessary, yielding a superior trade-off between accuracy and computation. Extensive evaluations across 11 benchmarks demonstrate the effectiveness and generalizability of SGL, achieving up to 91% pruning ratio for visual tokens while retaining competitive performance. The code will be publicly available at <https://github.com/NUS-HPC-AI-Lab/SGL>.*

1. Introduction

Building on the notable success of language models (LMs) [3, 23, 53, 59], vision-language models (VLMs) have become a focal point of research. Most current VLMs [7, 31, 32, 54, 71] integrate visual tokens from a vision encoder alongside textual tokens within an LM. However, such integration introduces significant inference overhead due to the sheer volume of visual tokens.

Visual token compression presents a compelling solution to improving the inference efficiency of VLMs. Recent works [28, 35, 60] aim to condense the information in visual tokens into fewer tokens or parameters, but generally require training, introducing additional overhead. Training-free alternatives, such as token merging in visual encoders [2, 48], offer a lighter solution but may overlook essential vision-language interactions. To bridge this gap, [5, 67] employs *partial information* from the LM e.g. attention maps from specific layers, to prune less important visual tokens. These approaches can ideally integrate seamlessly with existing VLMs without fine-tuning, showing promising effectiveness.

But how effective is visual token pruning across varying retention levels? To investigate this, we conduct an empirical study using FastV [5], a representative method that assesses visual token importance based on a single-layer attention map from the LM. For comparison, we also include an oracle method aggregating attention maps from all LM layers. As shown in Figure 1 (a), FastV struggles to maintain accuracy when the token retention ratio falls below 35%, whereas the oracle method remains competitive even with only 9% of visual tokens retained. This highlights that *the global information from the aggregated attention map accurately identifies essential visual tokens for VLM prediction*.

However, retrieving attention maps from all layers requires a full inference pass. One cannot obtain a precise assessment of token importance before inference. This accounts for FastV using a single layer’s attention as a proxy. To accurately distinguish essential vision tokens with minimal computation, we innovatively resort to a small model’s

*Equal contribution. Work done during an internship at DAMO Academy, Alibaba Group. wangbo.zhao96@gmail.com

†Corresponding author.



Figure 1. The motivation of our SGL. (a) **A single-layer attention map is suboptimal compared to the global attention maps aggregated from all layers.** We take InternVL2 [7] of 2B and 26B as representative examples. FastV [5] prunes visual tokens using the attention map from a single-layer, whereas FastV-oracle employs the aggregated attention map across all layers during inference. This approach allows for precise pruning of less significant visual tokens, maintaining performance with only 9% of the tokens retained. (b) **The small VLM exhibits a token retention pattern similar to the 26B model, preserving essential visual tokens relevant to the answer, regardless of the answer correctness.** We drop 80% less significant visual tokens and adopt ■ to mark those tokens with high attention scores. Thumbnails employed in InternVL2 [7] are presented in the left corner. (c) **The performance gap between small and large VLM is minimal compared to their computation disparity.** The 2B model achieves competitive performance with significantly fewer FLOPs compared to the 26B one. This also validates our soundness of using a small model to guide early exiting and token pruning in the large one.

global attention maps aggregated from all layers. As shown in Figure 1 (b), the token retention pattern derived from the small VLM closely mirrors that of the large VLM and consistently preserves tokens relevant to the answer, *regardless of output correctness*. This suggests that the small VLM’s overall attention map serves as a more precise proxy to guide visual token pruning in large VLMs.

Based on the above findings, we propose a **training-free** method, **Small VLM Guidance for Large VLMs (SGL)**, consisting of two insightful techniques. First, along the token pruning line, we develop *Small VLM-Guided visual token Pruning (SGP)*. Given an input image and its text prompt (question), inference is first performed by a small VLM, in which a global attention map is aggregated from all layers. This global attention map enables the calculation of vision tokens’ importance scores based on their interactions with prompt and generated tokens. The scores are then used to rank the visual tokens. Subsequently, the ranking results provide a guidance for pruning less important visual tokens in the large VLM, significantly reducing computation while preserving essential information for a correct answer.

It can be easily observed that the additional small VLM introduces some computational overhead. Fortunately, we notice that the performance gap between small and large VLMs is relatively minimal compared to their computation disparity (Figure 1 (c)). In other words, most “easy” questions could be correctly answered by the small VLM. This observation prompts us to make full use of the computation spent by the small VLM. To this end, we introduce *Small VLM Early Exiting (SEE)*. Specifically, after obtaining the small VLM’s prediction, we evaluate its decision score and directly terminate the inference pipeline without activating

the large VLM if the score exceeds the threshold. With the adoption of a small VLM, SEE serves an effective approach complementary to SGP: (i) The computation of the large VLM could be completely skipped for many “easy” questions (SEE); (ii) When the large VLM is activated by demand, most unimportant visual tokens can be pruned based on the small VLM’s aggregated attention map (SGP).

We demonstrate the effectiveness of SGL across 11 benchmarks, achieving up to 91% pruning of visual tokens in VLMs such as InternVL2 [7] in model sizes from 26B to 76B, while maintaining competitive performance. Moreover, our method can be seamlessly integrated with other VLMs *e.g.* Qwen-VL [54] and LLaVa-OV [27], highlighting its versatility in enhancing VLM efficiency across model architectures.

2. Related Works

VLMs. Advancements in language models (LMs) [3, 23, 53, 59] have driven significant progress in vision-language models (VLMs) [7, 31, 32, 54, 71]. Most VLMs use a visual encoder, such as ViT [10], to extract visual tokens connected to an LM via a projection layer, significantly increasing token length and leading to high computational and memory demands. For instance, LLaVa [32] processes 576 tokens for a 336×336 image resolution, while an enhanced version [31] processes 2304 tokens at higher resolutions. InternVL2 [7] introduces up to 10,496 tokens through dynamic high-resolution techniques, and video understanding models [30, 58, 68] handle thousands of tokens across frames.

Our method addresses the token overhead by using a small VLM to guide token reduction in larger VLMs, seam-

lessly applying to such models.

Visual token compression. Compressing visual tokens is a promising approach to reduce computational and memory costs in transformer-based vision models. Methods such as token pruning [29, 46], token merging [2, 6], and token skipping [18, 21, 40, 69, 70] have been extensively studied for tasks like image classification and segmentation. In VLMs, methods like Q-Former [28], token distillation [60], and parameter alignment [35] compress visual tokens but often require additional training. Training-free techniques [2, 5, 48, 67] merge or prune tokens based on LM attention maps, but approaches that merge tokens in the visual encoder [2, 48] may miss key vision-language interactions. Other methods [5, 67] prune tokens using attention maps from specific LM layers, which may fail to accurately capture essential tokens at low retention levels.

In comparison, this work proposes leveraging the aggregated attention map across all layers of a small VLM to comprehensively rank token importance, guiding more effective pruning in a larger VLM.

Model uncertainty/confidence estimation is essential for reliable predictions [9, 11, 15, 19, 20, 34, 42, 65]. Recent work on large models focuses on estimating the confidence of LM-generated text through information-based [13, 25], ensemble-based [36], density-based [47, 61], and reflexivity-based [24] methods. High-uncertainty content is often reviewed or processed by more advanced models for increased reliability [17, 64]. We recommend [12, 16] for a more comprehensive review.

Our SEE uses the predictions of the small VLM by measuring its confidence to determine when to activate the larger VLM, enhancing the trade-off between performance and efficiency. In addition, a consistency criterion is proposed to facilitate the early-exiting decision making procedure.

3. Small Guides Large (SGL) in VLMs

We first provide the preliminaries of VLMs in Section 3.1. Then, the proposed **training-free** Small VLM-Guided visual token Pruning (SGP) and Small VLM Early Exiting (SEE) techniques are detailed in Sections 3.2 and 3.3, respectively.

3.1. Preliminary of Vision-language Models

VLMs [7, 31, 32, 54, 71] primarily follow a framework where a vision encoder [10, 45, 51] converts an image into a sequence of visual tokens. These tokens are then combined with textual prompt tokens and fed into a language model [23, 52, 53, 59] to generate responses.

Specifically, an input image \mathbf{I} is encoded into visual tokens by a vision encoder model VM :

$$\mathbf{x}_I = VM(\mathbf{I}) \in \mathbb{R}^{N_I \times C}, \quad (1)$$

where N_I represents the image token number. The associated textual prompts are tokenized into prompt tokens

$\mathbf{x}_T \in \mathbb{R}^{N_T \times C}$. As observed in [5], we usually have $N_I \gg N_T$, especially for high-resolution images. The tokens \mathbf{x}_I and \mathbf{x}_T are concatenated and fed into a language model LM to generate responses auto-regressively:

$$\mathbf{p}_G^i = LM(\mathbf{x}_I, \mathbf{x}_T, \mathbf{x}_G^{1:i-1}) \in \mathbb{R}^{C_T}, \quad (2)$$

where \mathbf{p}_G^i denotes the probability distribution over the vocabulary of size C_T . The previous generated tokens $\mathbf{x}_G^{1:i-1}$ are used to predict the next token. The probability \mathbf{p}_G^i is converted into the token embedding \mathbf{x}_G^i via sampling, such as the argmax operation.

3.2. Small VLM-Guided Visual Token Pruning

The inference efficiency of VLMs is greatly impacted by the large number of vision tokens. A promising approach to mitigate this involves pruning less essential visual tokens using attention maps. However, pruning based on a single-layer attention map, as in [5], falls short compared to using an *oracle attention map aggregated from all layers* (Figure 1 (a)). Yet, obtaining this oracle attention map requires a full, computationally costly inference pass, making it impractical for real-world use. The key challenge is thus:

How can we efficiently acquire a precise attention map for effective visual token pruning?

Based on our findings in Figure 1(b) that the aggregated attention map closely resemble that of a large VLM, we introduce SGP (Figure 2(a)): using the *small VLM’s aggregated attention map* as an efficient and precise proxy to guide visual token pruning in a large VLM.

Aggregating attention maps in the small VLM. We initiate inference with a compact vision-language model, VLM^S (*e.g.* InternVL2-2B [7]), comprising a small vision model, VM^S , and a small language model, LM^S . This reduced model size significantly cuts computational costs compared to larger VLMs. We input visual tokens $\mathbf{x}_I \in \mathbb{R}^{N_I \times C}$ from vision model VM^S and textual prompt tokens $\mathbf{x}_T \in \mathbb{R}^{N_T \times C}$ into LM^S to generate answers $\mathbf{x}_G \in \mathbb{R}^{N_G \times C}$, where N_G denotes the number of generated tokens.

The inference process of LM^S involves a pre-filling stage followed by a decoding stage. We update two attention maps A^P, A^D for the two stages, respectively.

(i) *Pre-filling.* In this stage, attention maps are extracted from each layer and head, denoted as $A_{j,k}^P \in \mathbb{R}^{(N_I+N_T) \times (N_I+N_T)}$, where j and k denote the layer and head index, respectively. Due to the causal nature of attention in LM^S , $A_{j,k}^P$ is a lower triangular matrix. We specifically focus on the attention scores that visual tokens receive from prompt tokens. Therefore, we retrieve the bottom-left block

$$A_{j,k}^P \in \mathbb{R}^{(N_I+N_T) \times (N_I+N_T)} \Rightarrow \tilde{A}_{j,k}^P \in \mathbb{R}^{N_T \times N_I}. \quad (3)$$

We then sum up the N_T scores for each image token in $\tilde{A}_{j,k}^P$, producing $\tilde{A}_{j,k}^P \in \mathbb{R}^{N_I}$. During the forward pass, these

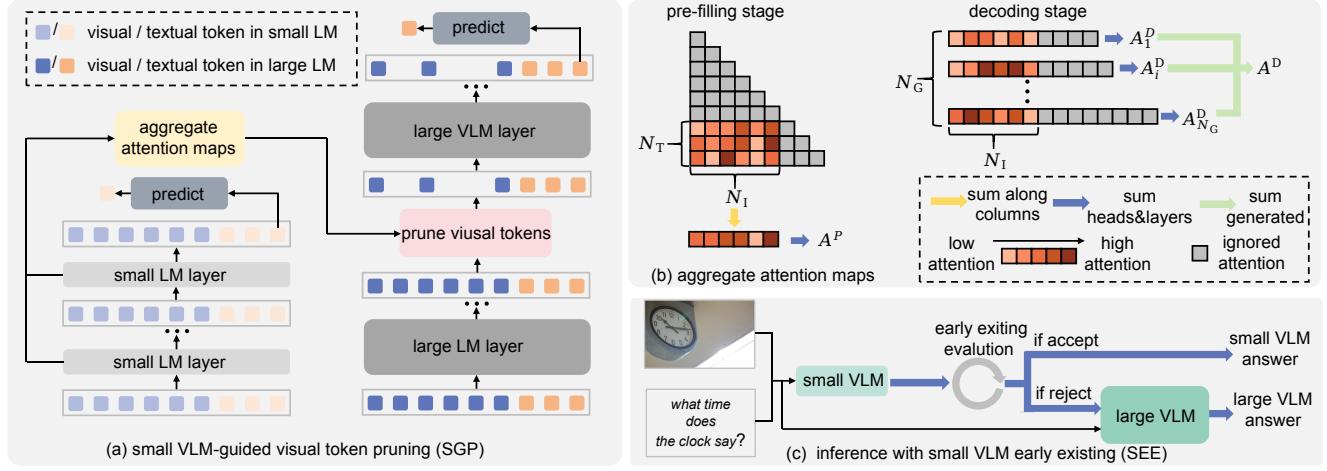


Figure 2. Overview of SGL. (a) **Small VLM-guided visual token pruning in a large VLM (SGP).** We update a global attention map aggregated from all layer of a small VLM. This global attention map is used to rank visual tokens and guide the visual token pruning in a large VLM. (b) **Aggregation of attention maps in SGP.** We aggregate the attention score of visual tokens received from prompt tokens and generated tokens across all heads and layers in the small LM. Higher scores indicate greater significance. (c) **Inference with Small VLM Early Exiting (SEE).** When the early exiting decision score from the small VLM is sufficient, the larger VLM will not be invoked.

attention maps are aggregated across layers and heads:

$$\mathbf{A}^P = \sum_{j=1}^L \sum_{k=1}^H \bar{\mathbf{A}}_{j,k}^P, \quad (4)$$

where L and H denote the number of layers and heads in LM_S , respectively. Note that we progressively update \mathbf{A}^P in an *accumulative* manner without caching all $\mathbf{A}_{j,k}^P$. This procedure is illustrated in Figure 2(b), left.

(ii) *Decoding.* Attention scores of N_I visual tokens from the i -th generated token can be denoted as $\mathbf{A}_{i,j,k}^D \in \mathbb{R}^{N_I}$ for head- k in layer- j . These scores are accumulated as

$$\mathbf{A}^D = \sum_{i=1}^{N_G} \sum_{j=1}^L \sum_{k=1}^H \mathbf{A}_{i,j,k}^D. \quad (5)$$

The aggregation in the decoding phase is visualized in Figure 2(b), right. Upon inference completion, we calculate the overall attention scores for vision tokens via $\mathbf{A} = \mathbf{A}^P + \mathbf{A}^D$. This comprehensive assessment \mathbf{A} is further employed to rank and prune visual tokens.

Visual token pruning in the large VLM. To improve the efficiency of a larger model VLM^L (*e.g.* InternVL2-26B [7]), we prune less important visual tokens based on the ranking obtained from \mathbf{A} . Specifically, the same image is fed into its vision model VM^L , producing visual tokens. Since VLM^S and VLM^L share the same architecture suite, VM^L outputs the same number of visual tokens. These tokens, combined with prompt tokens, are fed into LM^L . Inspired by FastV [5], we retain only the top $R\%$ of important visual tokens in an intermediate layer of LM^L , as determined by the ranking. Owing to the comprehensive importance score from the

small VLM, we can apply a low retention ratio (*e.g.* 5%) at an early layer (*e.g.*, the 2-nd layer), significantly reducing the computational cost of VM^L .

3.3. Small VLM Early Exiting

While SGL effectively reduces the token load in the large VLM, incorporating a small VLM does add some overhead relative to using the large VLM alone. Fortunately, the performance gap between small and large VLMs is relatively minor compared to their computational difference, indicating that the small VLM’s outputs are often quite competitive. This inspires us to devise Small VLM Early Exiting (SEE), maximizing the utility of the small VLM by assessing its outputs. For some “easy” questions, the inference pipeline exits early after obtaining the small VLM’s prediction, further enhancing the inference efficiency. We demonstrate its pipeline in Figure 2 (c).

During small VLM inference, the token generation probability can be recorded to estimate the answer confidence. A straightforward yet effective method for confidence estimation involves calculating the length-normalized sequence probability [12, 41], which can be expressed as:

$$\mathcal{S}_{\text{confidence}} = \exp \left\{ \frac{1}{N_G} \log P(\mathbf{x}_G^1, \dots, \mathbf{x}_G^{N_G}) \right\}, \quad (6)$$

where

$$P(\mathbf{x}_G^1, \dots, \mathbf{x}_G^{N_G}) = \prod_{i=1}^{N_G} P(\mathbf{x}_G^i | \text{LM}^S(\mathbf{x}_I, \mathbf{x}_T, \mathbf{x}_G^{1:i-1})). \quad (7)$$

In our token pruning scenario, apart from the naive confidence metric $\mathcal{S}_{\text{confidence}}$, we further propose a **consistency**

method	token ratio	visual question answering				comprehensive benchmark				visual grounding			score ratio
		TextVQA	ChartQA	DocVQA	GQA	SEED	MMBench	MM-Vet	MME	RC	RC+	RC-g	
InternVL2-26B [7]	100%	82.45	84.92	92.14	64.89	76.78	83.46	64.00	2270	91.24	86.67	88.44	100.00%
InternVL2-2B [7]	100%	73.19	76.24	85.93	61.16	71.62	72.93	43.30	1878	82.26	73.53	77.55	87.25%
26B w/ ToMe [2]	64%	80.22	76.24	79.51	64.49	75.60	82.74	60.10	2235	84.02	78.91	80.35	94.24%
	35%	75.74	62.44	66.79	63.61	73.84	81.28	52.50	2178	71.08	64.97	68.08	85.20%
	9%	51.69	28.60	28.46	57.52	65.19	73.09	37.70	1933	20.33	17.74	19.36	54.28%
26B w/ FastV [5]	64%	82.26	85.08	92.20	64.80	76.81	83.24	63.20	2270	91.30	86.66	88.30	99.84%
	35%	75.62	71.68	68.32	61.20	71.64	78.31	45.00	2140	85.06	77.61	81.39	88.28%
	9%	43.84	26.20	26.81	44.90	54.56	62.33	31.60	1799	19.65	16.66	17.22	46.99%
26B w/ SGP (ours)	64%	82.41	85.04	92.12	65.07	76.71	83.30	65.60	2259	91.07	86.71	88.05	100.14%
	35%	81.97	81.68	91.14	64.62	75.72	82.17	63.20	2258	89.38	84.35	86.07	98.36%
	9%	78.98	72.96	87.26	62.10	72.23	75.56	52.10	2004	80.36	72.22	77.45	89.58%

Table 1. **Comparison between SGP and previous visual token pruning methods.** “Token ratio” denotes the average ratio of retrained visual tokens. “26B” denotes the original InternVL2-26B. In “26B w/ SGP (ours)”, we employ the aggregated attention map across all layers in InternVL2-2B to guide the visual token pruning in InternVL2-26B. For fair comparison, we do not employ SEE in these experiments. The “score ratio” is obtained by calculating the ratio of each score relative to InternVL2-26B, followed by averaging these ratios.

score for making early-exiting decisions. Specifically, we can naturally hypothesize that the small VLM accurately identifies essential visual tokens when it provides a correct answer. Conversely, if the small VLM’s answer is correct, a consistent prediction should be obtained when visual tokens are pruned by SGP (Section 3.2). In this basis, we introduce a consistency score $S_{\text{consistency}}$ to measure the consistency of the generation after visual token pruning. A higher score indicates a higher probability that the small VLM yields a correct answer, where early exiting is more reliable. Let $\text{LM}^{S'}$ represent the small language model with pruned visual tokens, the consistency score is obtained by

$$S_{\text{consistency}} = \prod_{i=1}^{N_G} P\left(\mathbf{x}_G^i \mid \text{LM}^{S'}(\mathbf{x}_I, \mathbf{x}_T, \mathbf{x}_G^{1:i-1})\right). \quad (8)$$

It is worth noting that the calculation of $S_{\text{consistency}}$ is extremely efficient, because: (i) in Equation 8, visual tokens \mathbf{x}_I , text tokens \mathbf{x}_T and the generated tokens \mathbf{x}_G have all been obtained in Section 3.2, thus the consistency score can be computed in parallel rather than autoregressively; (ii) a high pruning ratio significantly reduces computational cost. We find that removing 95% visual tokens is feasible in practice. In this scenario, calculating $S_{\text{consistency}}$ requires <10% of the initial inference time with the small VLM. Finally, we compute the final early-exiting **decision score**:

$$S = \frac{1}{2}(S_{\text{confidence}} + S_{\text{consistency}}). \quad (9)$$

The inference pipeline exits early at the small VLM when the score is above a predefined threshold. In Section 4, we empirically show that our early-exiting criterion S outperforms other early-exiting criteria, such as quantile [17], entropy [12], and either one item $S_{\text{consistency}}$ or $S_{\text{confidence}}$.

To sum up, the small VLM in our SGL plays two roles. For any input, it first performs inference, producing

(i) The importance scores \mathbf{A} for vision tokens (SGP).

(ii) An early prediction and the corresponding early-exiting decision score S (SEE);

If the large VLM is decided to be activated by SEE, SGP prunes a large amount of unimportant visual tokens to accelerate the inference of the large VLM.

4. Experiment

4.1. Experimental Setup

Models. We conduct experiments using InternVL2 [7], which provides checkpoints for various model sizes, facilitating the exploration of small VLMs in guiding visual token pruning in large VLMs. Specifically, we use InternVL2-2B as the small VLM and InternV2L-26B as the large VLM by default. We also include Qwen-VL [54] and LLaVa-OV [27] to evaluate the generalizability of our method. The default setting in experiments is marked in color.

Evaluation benchmarks. We conduct experiments on four VQA benchmarks including TextVQA [50], ChartQA [38], DocVQA [39], and GQA [22]. To evaluate performance in visual grounding, we introduce RefCOCO (RC) [62], RefCOCO+ (RC+) [62], and RefCOCOg (RC-g) [37]. Additionally, we assess the model’s capability in general multimodal understanding on comprehensive benchmarks such as SEED[26], MMBench [33], MM-Vet [63], and MME [14].

4.2. Comparing SGP with Previous Methods

We first validate the effectiveness of our SGP without the early-exiting mechanism. The comparison with representative visual token compression methods, including ToMe [2] and FastV [5], is presented in Table 1, across different average visual token retention ratios. All experiments are conducted based on InternVL2-26B model, consisting of 48 layers. For our method and FastV [5], we prune 60%,

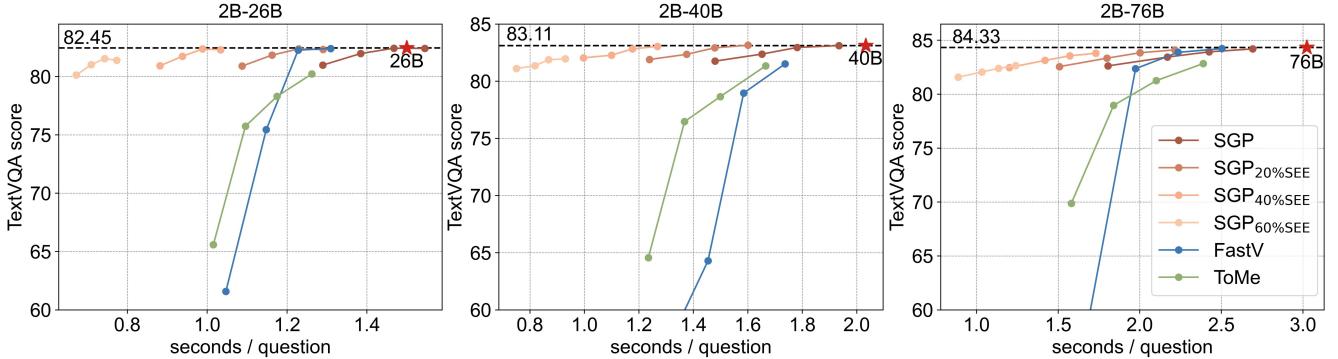


Figure 3. **Performance-efficiency curves of SGL (SGP + SEE).** The results with 18%, 35%, 50%, and 64% visual token retention ratios are presented as a curve. For the 26B and 40B, we use an NVIDIA H20 GPU, and the 76B is sharded on two GPUs.

80% and 95% visual tokens at the 19-th, 9-th, and 2-th layer, achieving average token retention ratios of 64%, 35%, and 9%, respectively. ToMe [2] performs token merging prior to the language model, with the merging ratio adjusted to achieve similar average token retention ratios.

At a relatively high token retention ratio, such as 64%, all methods exhibit competitive performance across various tasks. This suggests significant visual token redundancy in VLMs, underscoring the importance of visual token pruning.

When the token retention ratio is decreased to 35%, the performance of ToMe and FastV starts to drop, particularly in OCR-related tasks, including TextVQA [50], ChartQA [38], and DocVQA [39] as well as visual grounding tasks. Their performance on MM-Vet [63] also drops significantly, since it also includes many OCR-related questions. These tasks require methods to accurately retain answer-related visual tokens to understand image details. This performance decline demonstrates that ToMe and FastV can not accurately retain essential tokens. In contrast, our method maintains competitive performance across all tasks.

With only 9% of visual tokens retained, FastV and ToMe collapse across all tasks, as critical visual tokens are lost due to inaccurate pruning. In this challenging scenario, our method experiences only a marginal performance drop compared to other methods, achieving over 89% of the original InternVL2-26B’s performance. The visualization in Figure 5 also validates the superiority of our SGP, which successfully preserves the tokens most relevant to a correct answer, thanks to the global attention maps aggregated from the small VLM.

4.3. SGP with SEE Towards Improved Efficiency

We further validate the superiority of our SGL by incorporating both SGP and SEE mechanisms. The performance-efficiency curves for varying token retention ratios (18%, 35%, 50%, and 64%) across large VLMs of different sizes (InternVL2-{26B, 40B, 76B}) on TextVQA are presented in Figure 3. As discussed in Section 3.3, we perform early exiting based the decision score of the small VLM’s answers to reduce the invocation of the large VLM. When the large

attention map source	TextVQA	SEED	RC	score ratio
all layers of 26B (oracle)	80.04	71.90	84.49	94.44%
one layer of 26B (FastV [5])	43.84	54.56	20.33	48.37%
10% layers of 2B	44.40	63.03	18.09	51.92%
30% layers of 2B	57.42	63.07	15.20	56.15%
50% layers of 2B	74.16	68.17	56.74	80.31%
70% layers of 2B	77.29	70.96	80.33	91.40%
all layers of 2B (ours)	78.98	72.23	80.36	92.64%

Table 2. **Performance with attention maps from different sources.** The visual token retention ratio is set to 9% for all experiments. Aggregating attention maps from all layers of the small model (2B) achieves performance comparable to the oracle.

VLM is activated, SGP is employed to reduce the visual token redundancy. Note that the ratio of early exiting can be flexibly controlled by adjusting the decision threshold, as a smaller threshold induces a lower ratio, *i.e.* less invocation for the large VLM. Here we present the performance curves at 60%, 40%, and 20% early exiting ratios, denoted as SGP_{60%SEE}, SGP_{40%SEE}, and SGP_{20%SEE}, respectively.

It can be observed that, with the 26B large VLM, our method SGP without SEE yields slower inference compared to FastV and ToMe, due to the overhead of the 2B small VLM. However, scaling the large VLM to 40B and 76B results in competitive inference speeds and superior performance relative to FastV and ToMe, particularly at low token retention ratios. Additionally, the proposed SEE enables SGP to maintain competitive performance at 20% and 40% early-exiting ratios while significantly reducing the average inference time across all VLM sizes. These results demonstrate the effectiveness of our SEE in identifying unreliable answers from the small VLM and appropriately invoking the large VLM. To summarize, our SGL offers superior trade-off between efficiency and performance.

4.4. Ablation Study of Key Designs

Effectiveness of all-layer attention maps. To verify the superiority of aggregating attention maps from all layers of the small VLM, we experiment using different sources to

used token	TextVQA	SEED	RC	score ratio
last prompt token	79.40	69.53	13.63	67.26%
prompt tokens	76.15	72.25	59.90	84.03%
generated tokens	79.38	63.47	83.51	90.16%
prompt + generated tokens	78.98	72.23	80.36	92.64%

Table 3. **Performance of using different tokens in visual token importance evaluation.** Prompt and generated tokens provide a comprehensive evaluation of visual tokens

guide visual token pruning. All experiments are conducted with a 9% retention ratio without SEE.

The results shown in Table 2 indicate that using attention maps aggregated from the small model outperforms using a single layer from the large model, such as FastV [5]. Moreover, the average performance consistently improves when the number of aggregated layers increases. This demonstrates that leveraging attention maps from multiple layers helps accurately retain essential visual tokens, achieving performance comparable to the oracle. Notably, our method even slightly outperforms the oracle on the SEED benchmark, highlighting its effectiveness.

Key tokens used for attention aggregation. Our SGP adopts both prompt and generated tokens in the attention maps to assess the importance of visual tokens. We ablate this design choice without SEE in Table 3. It can be observed that using only generated or prompt tokens might result in unstable performance. For example, generated tokens achieve the best performance on the RC dataset but perform poorly on the SEED benchmark. Similarly, using only the last prompt token, as in FastV [5], also leads to instability, particularly on the RC dataset. These results demonstrate that employing both prompt and generated tokens provide a comprehensive evaluation of visual tokens.

Different strategies used in SEE. We further investigate the effectiveness of the early-exiting criteria used in our SEE. The proposed strategy \mathcal{S} is compared with other strategies, including only length-normalized sequence probability ($\mathcal{S}_{\text{confidence}}$ in Equation 6) [12, 41], consistency score ($\mathcal{S}_{\text{consistency}}$ in Equation 8), quantile [17] and entropy [12]. For the quantile strategy, the top 75th, 50th, and 25th percentile probabilities among generated tokens are used as confidence scores, denoted as Quantile_{Q1}, Quantile_{Q2}, and Quantile_{Q3}, respectively. In the entropy strategy, we aggregate the entropy of each generated token, where higher entropy indicates lower confidence. The results are presented in Figure 4, where SGP is omitted in all experiments.

It is observed that $\mathcal{S}_{\text{confidence}}$ outperforms other baselines except for $\mathcal{S}_{\text{consistency}}$. Building on this, we develop our strategy \mathcal{S} by integrating both $\mathcal{S}_{\text{confidence}}$ and $\mathcal{S}_{\text{consistency}}$, achieving the outstanding performance. It is noteworthy that the calculation of $\mathcal{S}_{\text{consistency}}$ is efficient, consuming <1/10 of the initial small VLM inference time, as analysed in Section 3.3.

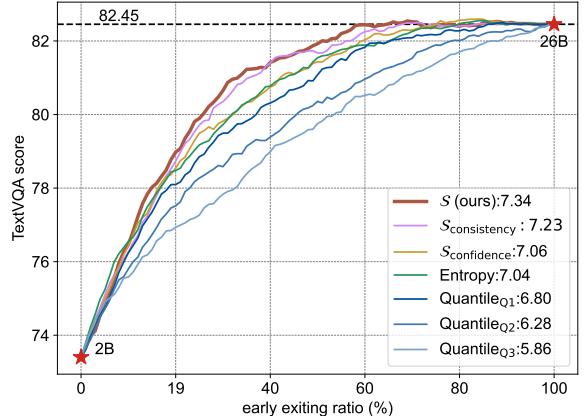


Figure 4. **Comparison of different early-existing decision scores.** We present the area between each strategy’s curve and the 2B model score alongside their names. A larger area indicates a more effective criterion. With the same early exiting ratio, a higher score reflects improved accuracy in identifying incorrect responses from the small VLM. Note that SGP is not adopted for clear comparison.

4.5. Visualization of Token Pruning and Answers

To provide a deeper understanding of our SGP, we visualize the token pruning results in Figure 5. The first row demonstrates that the small VLM helps retain essential visual tokens necessary for answering questions across various token retention ratios. The second row shows that, even when the answer is incorrect, the small VLM can still retain tokens related to the answer. This suggests that, although the small VLM lacks the precise reasoning and perception necessary for accurate answers, it possesses sufficient reasoning ability to identify target regions, thereby guiding visual token pruning in the large VLM. In the last row, we present a challenging example where the answer is located at the boundary of the image and appears in a small font size. In this difficult scenario, the small VLM produces a correct answer as the large VLM, verifying competitive performance of the small VLM. This enable us to conduct SEE using the responses from the small VLM during inference.

We also present the visual tokens pruning and answers from FastV [5]. We find that it can only preserve partial tokens relevant to the answer in the thumbnail and fails to precisely retain them in the main image. Consequently, this limitation impairs the model’s ability to perceive image details, leading to inaccurate predictions.

4.6. Generalization on Different Size Models

Small VLM. In Table 4, we evaluate the effectiveness of using small VLMs of varying sizes (InternVL2-{1B, 2B, 4B}) to guide visual token pruning in the larger InternVL2-26B model [7]. These models are constructed using different language models (LMs) from various sources. Specifically, InternVL2-1B uses Qwen2 [59], while InternVL2-2B and InternVL2-4B adopt InterLM2 [4] and Phi3 [1], respectively.



Figure 5. **Visualization of SGP under different visual token retention ratios and answers.** Visual tokens are pruned by 60%, 80%, and 95% at the 19th, 9th, and 2nd layers of the large VLM of 26B, which comprises 48 layers. This results in average token retention ratios of 64%, 35%, and 9%, respectively. Retained tokens are highlighted with ■. Thumbnails employed in InternVL are presented in the left corner.

small VLM size	LM source	TextVQA	SEED	RC	score ratio
1B	Qwen2 [59]	79.38	72.27	82.06	93.44%
2B	InterLM2 [4]	78.98	72.23	80.36	92.64%
4B	Phi3 [1]	79.70	73.65	75.39	91.73%

Table 4. **Performance of leveraging small VLMs of different sizes.** These small VLMs are based on various language models.

large VLM size	w/ ours	TextVQA	SEED	RC	score ratio
26B	✗	82.45	76.78	91.24	100.00%
26B	✓	78.98	72.23	80.36	92.64%
40B	✗	83.11	78.15	93.00	100.00%
40B	✓	79.96	74.11	79.99	92.38%
76B	✗	84.33	78.17	92.20	100.00%
76B	✓	80.72	73.93	81.82	92.98%

Table 5. **Visual token pruning for different-sized large VLMs.** The average retention ratio is set to 9%.

The results show that our method is robust to the choice of small model and maintains compatibility with different LMs. Surprisingly, InternVL2-1B performs slightly better than InternVL2-2B across three tasks, motivating further reducing the small VLM size in future studies.

Large VLM. We further substitute the large VLM in SGL by InternVL2-40B and InternVL2-76B, with the small VLM fixed as InternVL2-2B. The results in Table 5 indicate that our small model effectively guides significantly larger VLMs. This suggests that SGP is robust and has potential to guide the visual token pruning in huge VLMs.

method	token ratio	TextVQA	score ratio
Qwen2-VL-72B [54]	100%	85.50	100%
w/ SGP (ours)	64%	85.49	99.98%
w/ SGP (ours)	35%	85.13	99.56%
w/ SGP (ours)	9%	82.88	96.94%
LLaVa-OV-72B [27]	100%	79.30	100%
w/ SGP (ours)	64%	79.19	99.86%
w/ SGP (ours)	35%	78.65	99.18%
w/ SGP (ours)	9%	75.98	95.81%

Table 6. **Generalizability on Qwen2-VL and LLaVa-OV.** We adopt Qwen2-VL-2B and LLaVa-OV-0.5B to guide the visual token pruning in Qwen2-VL-72B and LLaVa-OV-0.5B, respectively.

4.7. Generalization on Various Architectures

We further assess the generalizability of SGL using Qwen2-VL [54] and LLaVa-OV [27]. The smallest models in the families, Qwen2-VL-2B and LLaVa-OV-0.5B, are used to guide visual token pruning in the largest models, Qwen2-VL-72B and LLaVa-OV-72B. The results in Table 6 reveal that SGL enables the large VLM to maintain approximately 96% of their original performance while achieving a retention ratio of 9% for visual tokens. This underscores the potential applicability of our method to varied VLM architectures.

5. Conclusion

In this study, we explore the effectiveness of attention maps for visual token pruning in VLMs. Our findings reveal that the attention map aggregated from all layers of a small VLM exhibits patterns akin to that of a larger VLM. Based on this insight, we introduce SGP, which prunes visual token in large VLMs under the guidance from a small VLM. This small VLM is further exploited to perform early exiting

(SEE) to make full use of its predictions. Both of these two techniques are training-free. Comprehensive experiments across 11 benchmarks demonstrate the method’s effectiveness, particularly at low visual token retention ratios.

Limitations and future works. Our method is primarily validated on multi-modal understanding tasks. Its application in recent VLMs [49, 55–57], unifying both understanding and generation, is worth studying in the future.

References

- [1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. [7](#), [8](#), [13](#)
- [2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022. [1](#), [3](#), [5](#), [6](#)
- [3] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. [1](#), [2](#)
- [4] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. [7](#), [8](#), [13](#)
- [5] 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. *arXiv preprint arXiv:2403.06764*, 2024. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [12](#)
- [6] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. In *ICCV*, pages 17164–17174, 2023. [3](#)
- [7] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. [1](#), [2](#), [3](#), [4](#), [5](#), [7](#), [13](#)
- [8] Zhe Chen, Jiannan Wu, Wenhui 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 *CVPR*, pages 24185–24198, 2024. [13](#)
- [9] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *ICCV*, pages 502–511, 2019. [3](#)
- [10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [2](#), [3](#), [13](#)
- [11] Stefan Eggemreich, Christian Payer, Martin Urschler, and Darko Štern. Variational inference and bayesian cnns for uncertainty estimation in multi-factorial bone age prediction. *arXiv preprint arXiv:2002.10819*, 2020. [3](#)
- [12] Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. Lm-polygraph: Uncertainty estimation for language models. *arXiv preprint arXiv:2311.07383*, 2023. [3](#), [4](#), [5](#), [7](#)
- [13] Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555, 2020. [3](#)
- [14] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. [5](#)
- [15] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023. [3](#)
- [16] Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, 2024. [3](#)
- [17] Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*, 2024. [3](#), [5](#), [7](#)
- [18] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *TPAMI*, 44(11):7436–7456, 2021. [3](#)
- [19] Yizeng Han, Yifan Pu, Zihang Lai, Chaofei Wang, Shiji Song, Junfeng Cao, Wenhui Huang, Chao Deng, and Gao Huang. Learning to weight samples for dynamic early-exiting networks. In *ECCV*, 2022. [3](#)
- [20] Yizeng Han, Dongchen Han, Zeyu Liu, Yulin Wang, Xuran Pan, Yifan Pu, Chao Deng, Junlan Feng, Shiji Song, and Gao Huang. Dynamic perceiver for efficient visual recognition. In *ICCV*, 2023. [3](#)
- [21] Yizeng Han, Zeyu Liu, Zhihang Yuan, Yifan Pu, Chaofei Wang, Shiji Song, and Gao Huang. Latency-aware unified dynamic networks for efficient image recognition. *TPAMI*, 2024. [3](#)
- [22] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. [5](#)
- [23] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. [1](#), [2](#), [3](#)
- [24] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac

- Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. 3
- [25] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023. 3
- [26] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 5, 12
- [27] 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 preprint arXiv:2408.03326*, 2024. 2, 5, 8, 13
- [28] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, pages 323–340. Springer, 2025. 1, 3
- [29] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022. 3
- [30] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2
- [31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 2, 3
- [32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2, 3
- [33] 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 *ECCV*, pages 216–233. Springer, 2025. 5
- [34] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020. 3
- [35] Feipeng Ma, Hongwei Xue, Guangting Wang, Yizhou Zhou, Fengyun Rao, Shilin Yan, Yueyi Zhang, Siying Wu, Mike Zheng Shou, and Xiaoyan Sun. Visual perception by large language model’s weights. *arXiv preprint arXiv:2405.20339*, 2024. 1, 3
- [36] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020. 3
- [37] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camбуру, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 5
- [38] 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. 5, 6
- [39] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, pages 2200–2209, 2021. 5, 6
- [40] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *CVPR*, pages 12309–12318, 2022. 3
- [41] Kenton Murray and David Chiang. Correcting length bias in neural machine translation. *arXiv preprint arXiv:1808.10006*, 2018. 4, 7
- [42] Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 59:101557, 2020. 3
- [43] NousResearch. Hermes-2-theta-llama-3-70b. <https://huggingface.co/NousResearch/Hermes-2-Theta-Llama-3-70B>, . 13
- [44] NousResearch. Nous-hermes-2-yi-34b. <https://huggingface.co/NousResearch/Nous-Hermes-2-Yi-34B>, . 13
- [45] 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 *ICML*, pages 8748–8763. PMLR, 2021. 3
- [46] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamiccvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 34: 13937–13949, 2021. 3
- [47] Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. Out-of-distribution detection and selective generation for conditional language models. In *ICLR*, 2022. 3
- [48] 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. 1, 3
- [49] Md Fahim Sikder, Resmi Ramachandranpillai, and Fredrik Heintz. Transfusion: generating long, high fidelity time series using diffusion models with transformers. *arXiv preprint arXiv:2307.12667*, 2023. 9
- [50] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, pages 8317–8326, 2019. 5, 6
- [51] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 3
- [52] InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities, 2023. 3
- [53] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambo, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2, 3
- [54] 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. 1, 2, 3, 5, 8, 13
- [55] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 9
- [56] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.
- [57] 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. *arXiv preprint arXiv:2408.12528*, 2024. 9
- [58] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 2
- [59] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 1, 2, 3, 7, 8, 13
- [60] Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, Ying Shan, and Yansong Tang. Voco-llama: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024. 1, 3
- [61] KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. Detection of word adversarial examples in text classification: Benchmark and baseline via robust density estimation. *arXiv preprint arXiv:2203.01677*, 2022. 3
- [62] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 5, 12
- [63] 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. *arXiv preprint arXiv:2308.02490*, 2023. 5, 6
- [64] Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint arXiv:2310.03094*, 2023. 3
- [65] Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. In *NeurIPS*, 2024. 3
- [66] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023. 13
- [67] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024. 1, 3
- [68] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llavanext: A strong zero-shot video understanding model, 2024. 2
- [69] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. *arXiv preprint arXiv:2410.03456*, 2024. 3
- [70] Wangbo Zhao, Jiasheng Tang, Yizeng Han, Yibing Song, Kai Wang, Gao Huang, Fan Wang, and Yang You. Dynamic tuning towards parameter and inference efficiency for vit adaptation. In *NeurIPS*, 2024. 3
- [71] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3

Appendix

A. SGP with SEE Towards Improved Efficiency

In Figure 6 and Figure 7, we further validate the superiority of our SGL by incorporating both SGP and SEE mechanisms, on SEED [26] and RefCOCO [62] benchmarks. It can be observed that, with the 26B large VLM, our method SGP without SEE yields slower inference compared to FastV and ToMe. This is attributed to the computational overhead of the 2B small VLM, particularly on RefCOCO, where it requires a non-negligible amount of time to auto-regressively generate a greater number of tokens compared to other datasets *e.g.* SEED. However, scaling the large VLM to 40B and 76B results in competitive inference speeds and superior performance relative to FastV and ToMe, particularly at low token retention ratios.

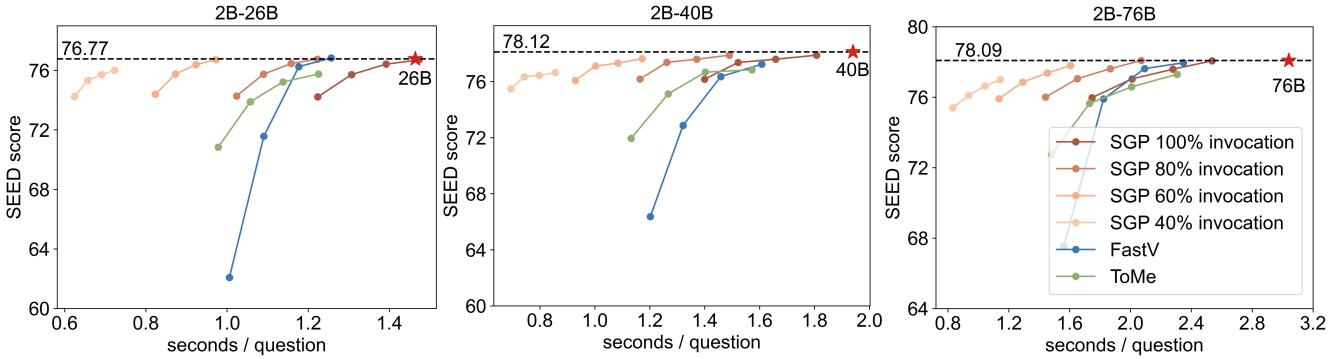


Figure 6. Performance-efficiency curves of SGL (SGP + SEE) on SEED [26]. The results with 18%, 35%, 50%, and 64% visual token retention ratios are presented as a curve. For the 26B and 40B, we use an NVIDIA H20 GPU, and the 76B is sharded on two GPUs.

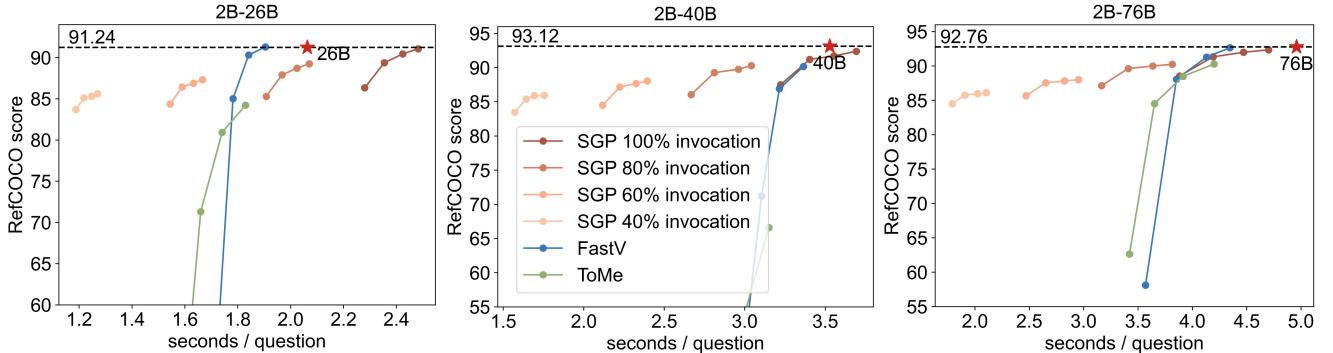


Figure 7. Performance-efficiency curves of SGL (SGP + SEE) on RefCOCO [62]. The results with 18%, 35%, 50%, and 64% visual token retention ratios are presented as a curve. For the 26B and 40B, we use an NVIDIA H20 GPU, and the 76B is sharded on two GPUs.

B. Memory Efficiency

In this section, we analyze the memory allocation of SGL. Our approach incorporates a small VLM *e.g.* InternVL2-2B in addition to the large VLM, which may introduce some additional memory overhead. Fortunately, the small VLM consumes only a minimal portion of memory compared to the large model. As a result, our method retains memory efficiency, as verified in Table 7.

C. Visualization

In Figure 8, we provide additional visualizations of examples where the small VLM (2B) fails to produce correct predictions, while the large VLM (26B), with visual tokens pruned by SGP, successfully predicts the correct answers. Notably, in these cases, the large VLM with FastV [5] also fails.

small VLM	small VLM memory	large VLM	large VLM peak memory	large VLM with SGL peak memory	Δ
2B	4.48 GiB	26B	51.60 GiB	54.24 GiB	+2.64GiB (5.11%)
2B	4.48 GiB	40B	77.94 GiB	80.60 GiB	+2.66GiB (3.41%)
2B	4.48 GiB	76B	147.64 GiB	147.25 GiB	-0.39 GiB (0.26%)

Table 7. **Mmeory analysis of SGL.** The meory of our method is measured with 9% average retention ratio. “small VLM memory” refers to the memory required to load the single small VLM. “Large VLM peak memory” represents the peak memory usage during inference with only the large VLM. “Large VLM with SGL peak memory” indicates the peak memory usage during inference of the large VLM when using the proposed SGL method (guided by a 2B model). Δ is defined as the difference between “Large VLM with SGL peak memory” and “Large VLM peak memory”. We report the ratio of Δ relative to “Large VLM peak memory”.

D. Model Descriptions

The configurations of InternVL [7], QWen2-VL [54], and LLaVa-OV [27] are comprehensively detailed in Tables 8, 9, and 10, respectively.

model name	language model	vision encoder	checkpoint
InternVL2-1B	Qwen2-0.5B [59]	InternViT-300M [8]	link
InternVL2-2B	InternLM2-chat-1.8B [4]	InternViT-300M [8]	link
InternVL2-4B	Phi-3-mini-128k-instruct [1]	InternViT-300M [8]	link
InternVL2-26B	InternLM2-chat-20B [4]	InternViT-6B [8]	link
InternVL2-40B	Nous-Hermes-2-Yi-34B [44]	InternViT-6B [8]	link
InternVL2-76B	Hermes-2-Theta-Llama-3-70B [43]	InternViT-6B [8]	link

Table 8. **Model descriptions of InternVL [7]**

model name	language model	vision encoder	checkpoint
Qwen2-VL-2B	Qwen2-1.5B [59]	ViT [10]	link
Qwen2-VL-76B	Qwen2-72B [59]	ViT [10]	link

Table 9. **Model descriptions of QWen2-VL [54]**

model name	language model	vision encoder	checkpoint
LLaVa-OV-0.5B	Qwen2-0.5B [59]	SigLIP [66]	link
LLaVa-OV-72B	Qwen2-72B [59]	SigLIP [66]	link

Table 10. **Model descriptions of LLaVa-OV [27].**

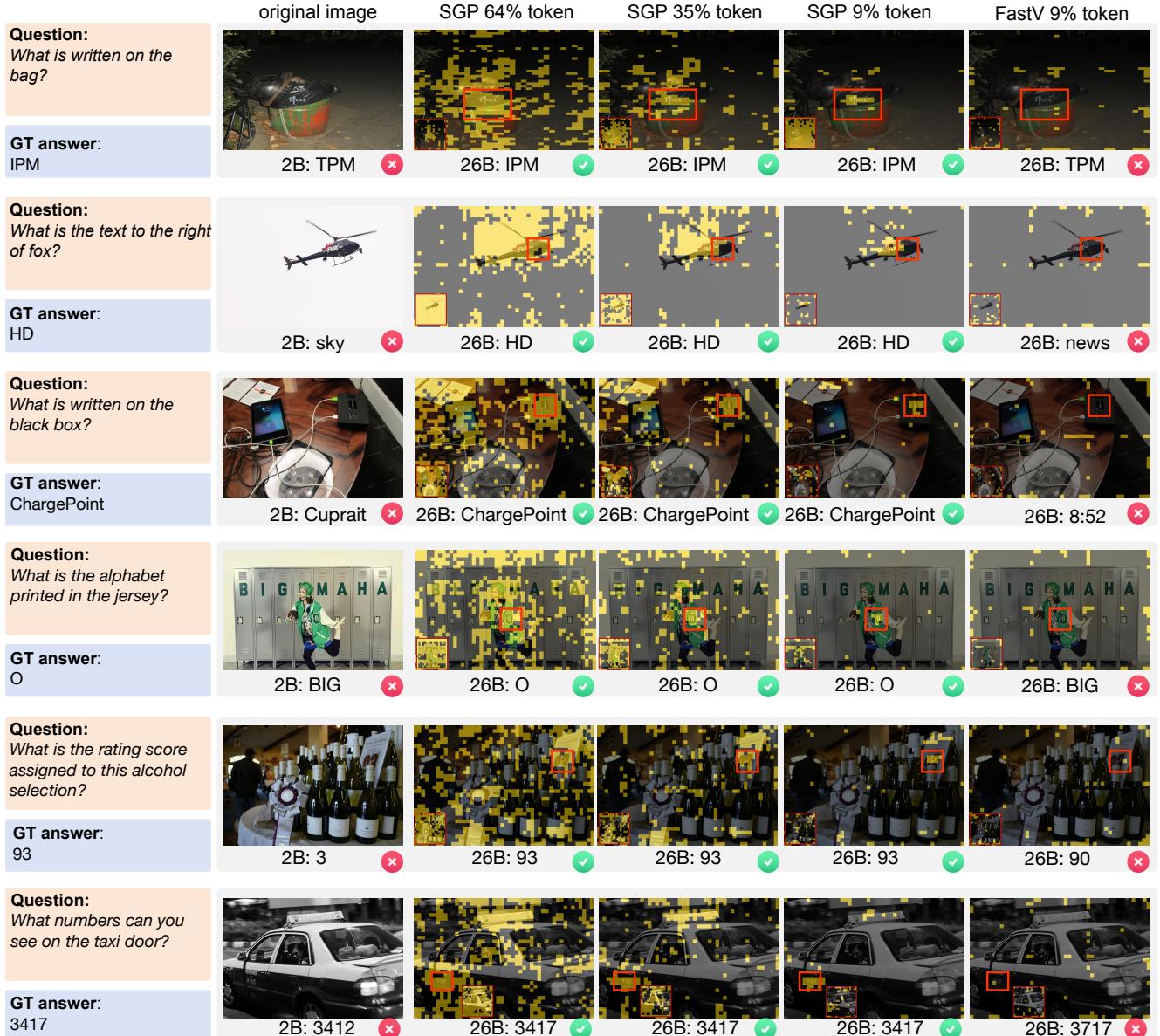


Figure 8. **Additional visualization of SGP under different visual token retention ratios and answers.** Visual tokens are pruned by 60%, 80%, and 95% at the 19th, 9th, and 2nd layers of the large VLM of 26B, which comprises 48 layers. This results in average token retention ratios of 64%, 35%, and 9%, respectively. Retained tokens are highlighted with ■. Thumbnails employed in InternVL are presented in the left corner.