

**Algorithm 1** Pseudo Code for Adaptive Token Prunings

---

```

1: Image features  $F$ , attention scores  $A$ , erank  $r$ , attention entropy  $H$ , maximum tokens  $T$  Selected
   token set  $S$ 
2: Normalize  $F$  and compute distance matrix  $D$ 
3:  $d_{len} \leftarrow f(H)$ ,  $\tau \leftarrow g(r)$ 
4: Sort tokens by descending  $A \rightarrow \pi$ 
5:  $S \leftarrow \emptyset$ ,  $M \leftarrow \text{all True}$ 
6: for each token  $i$  in  $\pi$  do
7:   if  $|S| \geq T$  then
8:     break
9:   end if
10:  if  $M[i] = \text{False}$  then
11:    continue
12:  end if
13:  Add  $i$  to  $S$ 
14:  if  $\text{position}(i) < d_{len}$  then
15:     $\theta \leftarrow \tau$  {adaptive threshold}
16:  else
17:     $\theta \leftarrow \text{base\_threshold}$ 
18:  end if
19:  Find neighbors  $C$  with  $D[i, j] \leq \theta$ 
20:  Suppress redundant tokens in  $C$  by updating  $M$ 
21: end for
22: return  $S$ 

```

---

**A PROPOSED METHOD DETAIL**

Algorithm 1 outlines the proposed adaptive token pruning strategy, which integrates attention-based importance with diversity-aware redundancy control. Given image features  $F \in \mathbb{R}^{N \times D}$  and [CLS] token attention score  $\alpha \in \mathbb{R}^N$  in the penultimate layer of the vision encoder, we first normalize features and compute the pairwise cosine distance matrix  $D \in \mathbb{R}^{N \times N}$ , defined as

$$D_{ij} = 1 - \frac{f_i \cdot f_j}{\|f_i\| \|f_j\|}. \quad (7)$$

Our strategy is governed by two parameters that adapt to image characteristics: a threshold length  $d_{len}$  derived from attention entropy  $H$ , and a similarity threshold  $\tau$  derived from erank  $r$ .

First, the similarity threshold  $\tau$  controls the aggressiveness of redundancy pruning. A high effective rank ( $r$ ) indicates that the token representations are already diverse. This implies we can safely employ a looser similarity criterion (a larger  $\tau$ ) to prune potential redundancies more aggressively without risking the loss of unique information. This behavior is modeled by a monotonic increasing function of  $r$ :

$$\tau(r) = \begin{cases} \tau_{\min}, & r \leq r_{\min}, \\ \alpha_\tau \ln(r) + \beta_\tau, & r_{\min} < r < r_{\max}, \\ \tau_{\max}, & r \geq r_{\max}. \end{cases} \quad (8)$$

Here,  $\tau_{\min}$  and  $\tau_{\max}$  are set to 80 and 100, respectively, while  $\alpha$  is 0.32 and  $\beta$  is -1.36.

Second,  $d_{len}$  defines the number of top-ranked tokens to which the adaptive threshold  $\tau$  is applied, while the remaining tokens are handled by a fixed base threshold. The rationale is that for high-attention entropy ( $H$ ) images, information is broadly distributed, so the special significance of the highest-attention tokens is diminished. We therefore apply a more consistent pruning strategy across all tokens by reducing  $d_{len}$  as  $H$  increases. This is captured by a monotonic decreasing function:

$$d_{len}(H) = \begin{cases} d_{\max}, & H \leq H_{\min}, \\ \alpha_d \ln(C - H) + \beta_d, & H_{\min} < H < H_{\max}, \\ d_{\min}, & H \geq H_{\max}. \end{cases} \quad (9)$$

Function	POPE	SQA	MME
Linear	84.30	68.34	1391
Exponential	84.26	68.77	1388
Logarithmic	<b>84.76</b>	<b>68.80</b>	<b>1400</b>

Table 7: Ablation study on the functional form of  $\tau(R)$ . Logarithmic mapping consistently outperforms linear and exponential alternatives.

Here,  $C$ ,  $H_{\min}$  and  $H_{\max}$  are set to 151, 4.5 and 5.0, respectively, while  $\alpha$  is 0.3 and  $\beta$  is 3.6.

Tokens are then sorted by attention scores in descending order, yielding a permutation  $\pi \in \{1, \dots, N\}^N$ . For each selected token, the algorithm applies either the adaptive threshold  $\tau$  (for the top  $d_{len}$  tokens) or a fixed base threshold to suppress neighboring tokens that are highly similar according to  $D$ . A binary mask  $\mathcal{M} \in \{0, 1\}^N$  is updated at each step to ensure redundancy removal. The process terminates when the maximum number of tokens  $T \in \mathbb{Z}$  is reached, and the algorithm returns the final selected set  $\mathcal{S} \subseteq \{1, \dots, N\}$  with  $|\mathcal{S}| \leq T$ .

#### A.1 ABLATION STUDY

In this section, we analyze the scoring function, a key component of our adaptive pruning framework, and present the empirical basis for our choice of a logarithmic function. An appropriate scoring function plays a crucial role in transforming the image complexity metric (erank) into a meaningful value that the model can effectively leverage.

To identify the optimal scoring technique, we transformed the image complexity metric using three representative functions—linear, exponential, and logarithmic—and compared their impact on model performance. As summarized in Table 8, our experimental results indicated that the logarithmic function achieved the best performance, followed by the linear and exponential approaches. This performance difference appears to stem from how each function handles the full spectrum of image complexity.

- A **linear function** tends to treat the difference between complexity scores of 10 and 20 the same as the difference between 100 and 110. However, for high-complexity images, it is often more ideal for the impact of score increases to diminish gradually. Therefore, a purely linear approach may not be optimal.
- An **exponential function**, on the other hand, can potentially amplify this issue. It risks exaggerating high complexity scores, which can generate outliers and lead to instability in the thresholding system, where a few complex images might dominate the outcome.
- In this context, a **logarithmic function** appears to be a more suitable approach for modeling this “diminishing returns” phenomenon. The logarithmic transformation tends to dampen the effect of high values, allowing for more stable thresholding, especially among images with very high complexity.

In summary, our experiments suggest that logarithmic scoring offers a more balanced approach. It provides sufficient sensitivity to distinguish between less complex images while mitigating the impact of extreme values from highly complex images, thereby ensuring greater system stability.

## B EFFICIENCY ANALYSIS

As shown in Table 8, the proposed method reduces FLOPs by **89%** under the 64-token setting, while still preserving **96.2%** of the original performance compared to the vanilla LLaVA-1.5-7B model. In particular, when compared with recent attention-based and diversity-based approaches such as VisPruner and DivPrune, the computational indicators (FLOPs, latency, GPU memory) remain nearly identical. However, in terms of accuracy, our method achieves the highest performance among the three.

All three methods share the property of performing pre-pruning before the LLM input, which substantially reduces the internal computation of the LLM. Among them, our method is especially

Method	Retain Tokens	FLOPs (T)	Latency (ms/sample)	GPU Memory (GB)	Accuracy
Vanilla (LLaVA-1.5-7B)	576	3.14	210	14.68	58.2
VisPruner (ICCV'25)	64	0.48	173	14.40	55.4
DivPrune (CVPR'25)	64	0.48	167	14.40	55.8
<b>Ours</b>	64	0.48	170	14.40	<b>56.0</b>

Table 8: Efficiency/accuracy comparison under identical evaluation protocol. Retain Tokens denotes the number of visual tokens kept after pruning (vanilla uses 576). All results are measured on the **TextVQA dataset** using a single NVIDIA A100 40GB GPU.

efficient since token selection relies solely on computing the entropy of layer attention values, introducing only minimal additional computational overhead.

In addition, our method is fully compatible with FlashAttention (Dao et al., 2022), enabling further efficiency gains when combined with state-of-the-art acceleration techniques. Overall, these results demonstrate that our method strikes an effective balance between computational efficiency and accuracy.

## C ADDITIONAL RESULTS

### C.1 EVALUATION ON OTHERS MODEL

In addition to LLaVA-1.5-7B, we also conducted experiments on larger models, namely LLaVA-1.5-13B (576 tokens) and LLaVA-NeXT-7B (2880 tokens). Across these settings, our method consistently demonstrated stable and strong performance, further validating the effectiveness of our approach (see Table 9 and Table 10).

### C.2 SUPPLEMENTARY EXAMPLES OF IMAGE COMPLEXITY-DEPENDENT PRUNING DIFFERENCES

As shown in Fig. 5, the qualitative patterns observed consistently reproduced across additional samples. For simple images (with low entropy and erank), attention-based pruning effectively captures the concentrated regions, while the additional benefits of diversity-based pruning are limited. Conversely, for complex images (with higher entropy and erank), diversity-based pruning ensures broader coverage, highlighting its strength in dispersed scenarios. These supplementary examples reinforce that image complexity is a key determinant of pruning effectiveness and motivate the need for an adaptive strategy that integrates both approaches.

### C.3 MORE EXAMPLES ON CHAIR

To further illustrate the differences between attention-based and diversity-based pruning in the image captioning task, we provide additional qualitative samples from the CHAIR dataset comparing FasterVLM as an attention-based method and DivPrune as a diversity-based method (Fig. 6 and Fig. 7). These cases illustrate how the two approaches differ in response style and hallucination tendency: DivPrune often yields broader and more descriptive captions but introduces hallucinated objects, whereas FasterVLM produces more conservative and focused descriptions.

Moreover, Fig.8 presents a controlled experiment where the token budget is fixed at 64 and the ratio between DivPrune- and attention-selected tokens (DivPrune-to-Attention ratio,  $R$ ) is varied in steps of 25%. We observe that the hallucinated objects frequently generated under pure DivPrune ( $R = 0$ ) gradually diminish as the share of attention-selected tokens increases, disappearing entirely when  $R \geq 50\%$ . In parallel, the response style evolves from speculative and exploratory to more factual and reliable as  $R$  increases, showing the stabilizing effect of attention-based token selection.

Method	VQA <sup>v2</sup>	GQA	VizWiz	SQA <sup>IMG</sup>	TextVQA	POPE	MME	MMB	MMB <sup>CN</sup>	Average
<i>Vanilla 576 Tokens</i>										
LLaVA-1.5-13B	80.0	63.3	53.6	72.8	61.2	86.0	1531	68.5	63.5	100%
<i>Retain 128 Tokens</i>										
FastV (ECCV'24)	75.3	58.3	54.6	74.2	58.6	75.5	1460	66.1	62.3	96.0%
PDrop (CVPR'25)	78.2	61.0	53.8	73.3	60.2	83.6	1489	67.5	62.8	98.4%
SparseVLM (ICML'25)	77.6	59.6	51.4	74.3	59.3	85.0	1488	68.4	62.6	97.8%
PruMerge+ (May'24)	76.2	58.3	52.8	73.3	56.1	82.7	1446	66.3	61.2	95.8%
VisionZip (CVPR'25)	76.8	57.9	52.3	73.8	58.9	82.7	1450	67.4	62.5	96.7%
DivPrune (CVPR'25)	77.1	59.2	53.5	72.8	58.0	86.8	1458	66.3	60.7	97.0%
Ours	77.5	58.7	53.0	73.9	58.9	86.3	1480	67.6	62.1	<b>97.8%</b>
<i>Retain 64 Tokens</i>										
FastV (ECCV'24)	65.3	51.9	53.8	73.1	53.4	56.9	1246	59.2	55.1	85.8%
PDrop (CVPR'25)	70.8	54.1	50.5	73.1	55.3	66.1	1247	63.1	56.6	88.7%
SparseVLM (ICML'25)	73.2	55.9	52.1	73.0	57.1	77.9	1374	65.2	60.3	93.5%
PruMerge+ (ICCV'25)	72.6	56.3	52.4	73.5	54.4	75.7	1338	65.0	59.3	92.3%
VisionZip (CVPR'25)	73.7	56.2	53.2	74.2	57.4	75.7	1380	64.9	61.3	93.9%
DivPrune (CVPR'25)	75.2	57.9	54.4	71.7	57.4	84.5	1454	64.1	59.8	95.6%
Ours	75.9	57.8	54.4	72.2	58.5	81.8	1433	65.7	61.7	<b>96.0%</b>

Table 9: **Results of different token pruning methods on 9 multimodal benchmarks.** Average is normalized to the full-token **LLaVA-1.5-13B** (set to 100%). MME is reported in its original score units, and it is included only in the *Perception* section to enable broader comparison with existing methods.

Method	VQA <sup>v2</sup>	GQA	VizWiz	SQA <sup>IMG</sup>	TextVQA	POPE	MME	MMB	MMB <sup>CN</sup>	Average
<i>Vanilla 2880 Tokens (Upper Bound)</i>										
LLaVA-NeXT-7B	81.3	62.5	55.2	67.5	60.3	86.8	1512	65.8	57.3	100%
<i>Retain 640 Tokens</i>										
FastV (ECCV'24)	77.0	58.9	53.9	67.4	58.1	79.5	1412	63.1	53.5	95.2%
PDrop (CVPR'25)	79.1	60.0	53.8	66.7	57.8	83.8	1475	64.1	55.2	97.0%
SparseVLM (ICML'25)	79.2	61.2	53.6	67.6	59.7	85.3	1456	65.9	58.6	98.8%
PruMerge+ (May'24)	78.2	60.8	57.9	67.8	54.9	85.3	1480	64.6	57.3	98.3%
VisionZip (CVPR'25)	79.1	61.2	57.1	68.1	59.9	86.0	1493	65.8	58.1	99.8%
DivPrune (CVPR'25)	79.3	61.9	55.7	67.8	57.0	86.9	1469	65.8	57.3	98.9%
Ours	79.3	61.9	56.3	69.0	59.7	86.3	1489	66.6	58.0	<b>100.0%</b>
<i>Retain 320 Tokens</i>										
FastV (ECCV'24)	61.5	49.8	51.3	66.6	52.2	49.5	1099	53.4	42.5	79.9%
PDrop (CVPR'25)	66.8	50.4	49.7	66.7	49.0	60.8	1171	55.5	44.7	82.5%
SparseVLM (ICML'25)	74.6	57.9	54.2	67.2	56.5	76.9	1386	63.1	56.7	94.6%
PruMerge+ (May'24)	75.3	58.8	57.7	68.1	54.0	79.5	1444	63.0	55.6	95.7%
VisionZip (CVPR'25)	76.2	58.9	56.2	67.5	58.8	82.3	1397	63.3	55.6	96.4%
DivPrune (CVPR'25)	77.2	61.1	55.6	67.7	56.2	84.7	1423	63.9	55.7	97.0%
Ours	77.8	60.3	55.9	67.8	59.1	84.2	1453	65.5	57.5	<b>98.3%</b>

Table 10: **Results of different token pruning methods on 9 multimodal benchmarks.** Average is normalized to the full-token **LLaVA-Next-7B** (set to 100%). MME is reported in its original score units, and it is included only in the *Perception* section to enable broader comparison with existing methods.

## D FARTHEST POINT SAMPLING

Farthest Point Sampling (FPS) is one of the simplest methods that guarantees diversity. Starting from an initial point, it iteratively selects the point that is farthest from the already chosen set by measuring the distance to the nearest selected point. For each point  $p_j$ , the minimum distance to the selected set  $S$  is defined as

$$d(p_j) = \min_{s \in S} \|p_j - s\|_2,$$

and the next point is chosen as

$$p_{i_t} = \arg \max_{p_j \in P \setminus S} d(p_j).$$



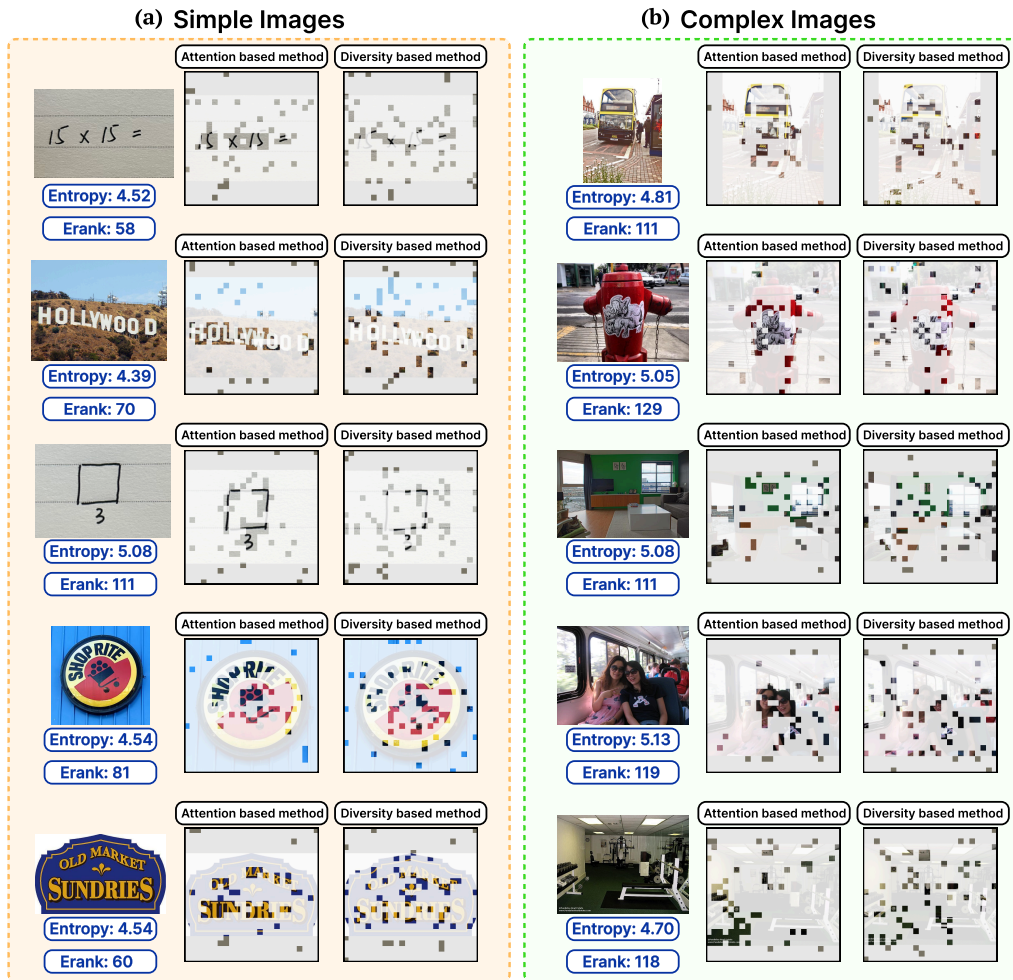


Figure 5: **Extended examples for simple vs. complex images.** The same trend as in Fig. 2 is observed: attention-based methods work well on simple images, while diversity-based methods cover complex images more broadly.

Repeating this process until the desired number  $k$  is reached ensures that the selected points are evenly distributed across the data space, providing a more balanced representation than simple random sampling.

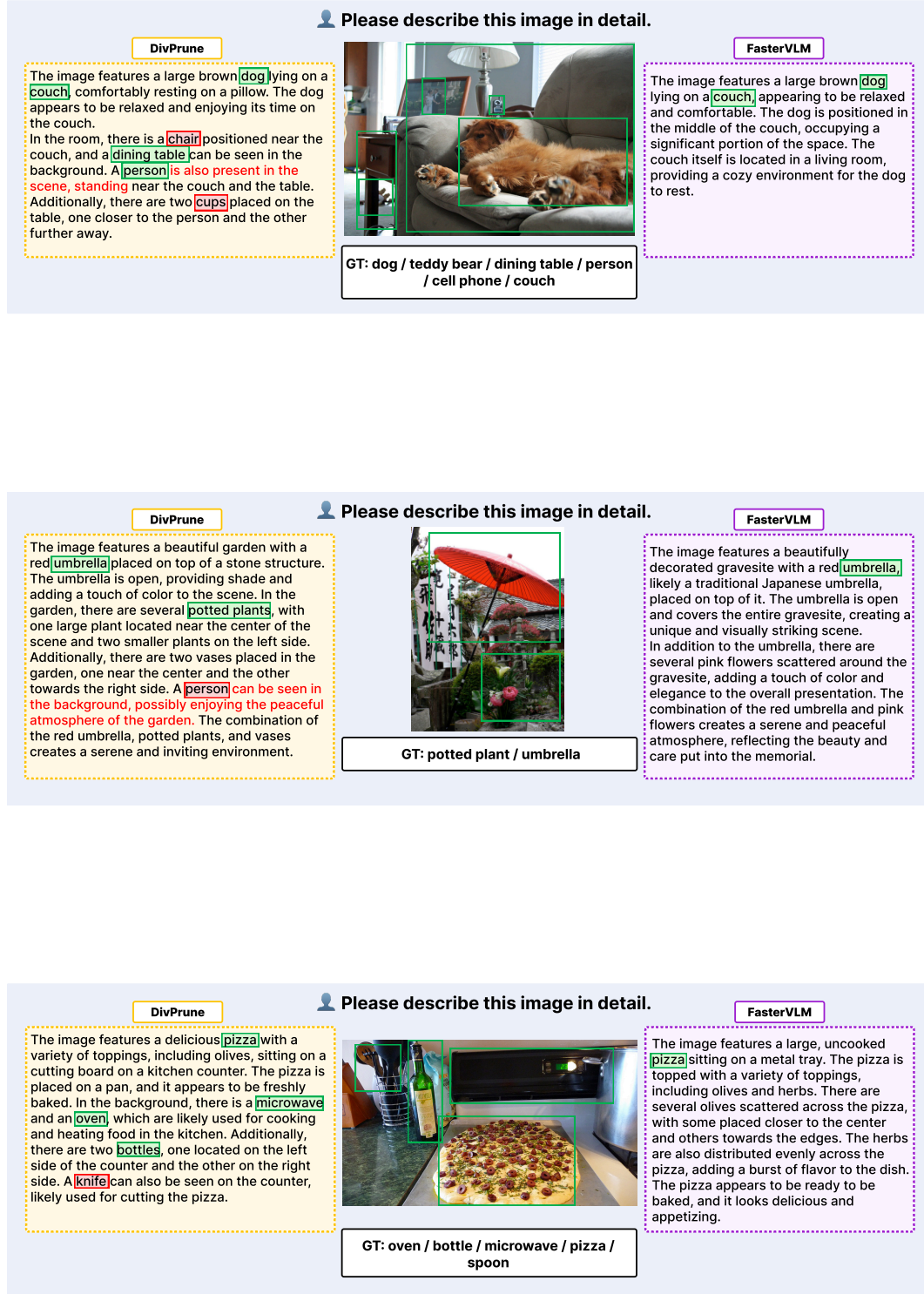


Figure 6: **CHAIR qualitative comparisons: FasterVLM vs. DivPrune (Set 1)**. In the annotations, ■ GT Obj. and ■ Hallucinated Obj. label object words; text indicates incorrect phrases.

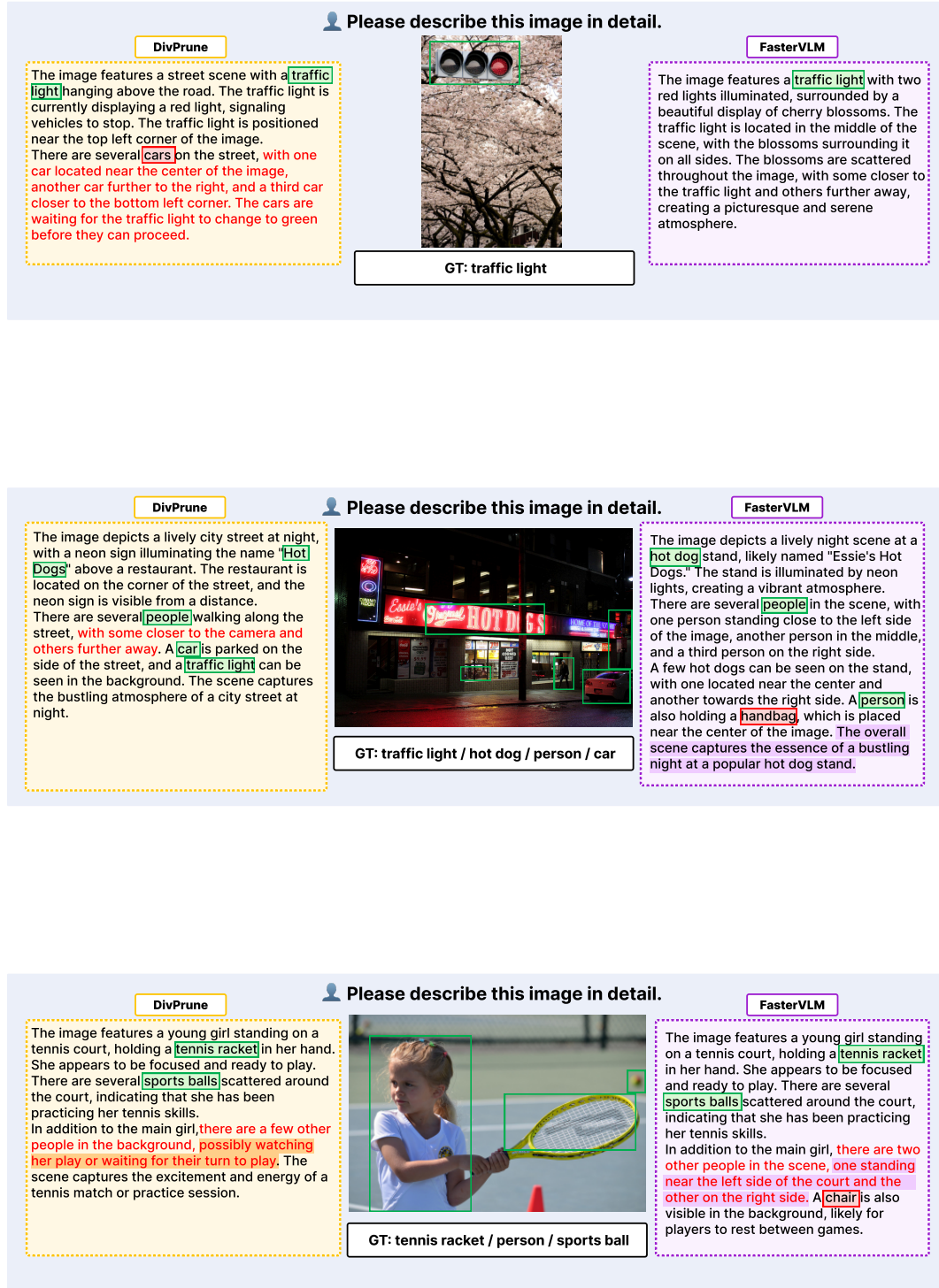


Figure 7: CHAIR qualitative comparisons: FasterVLM vs. DivPrune (Set 2). In the annotations, ■ GT Obj. and ■ Hallucinated Obj. label object words; ■ marks DivPrune’s phrasing; ■ marks FasterVLM’s phrasing; **red text** indicates incorrect phrases.

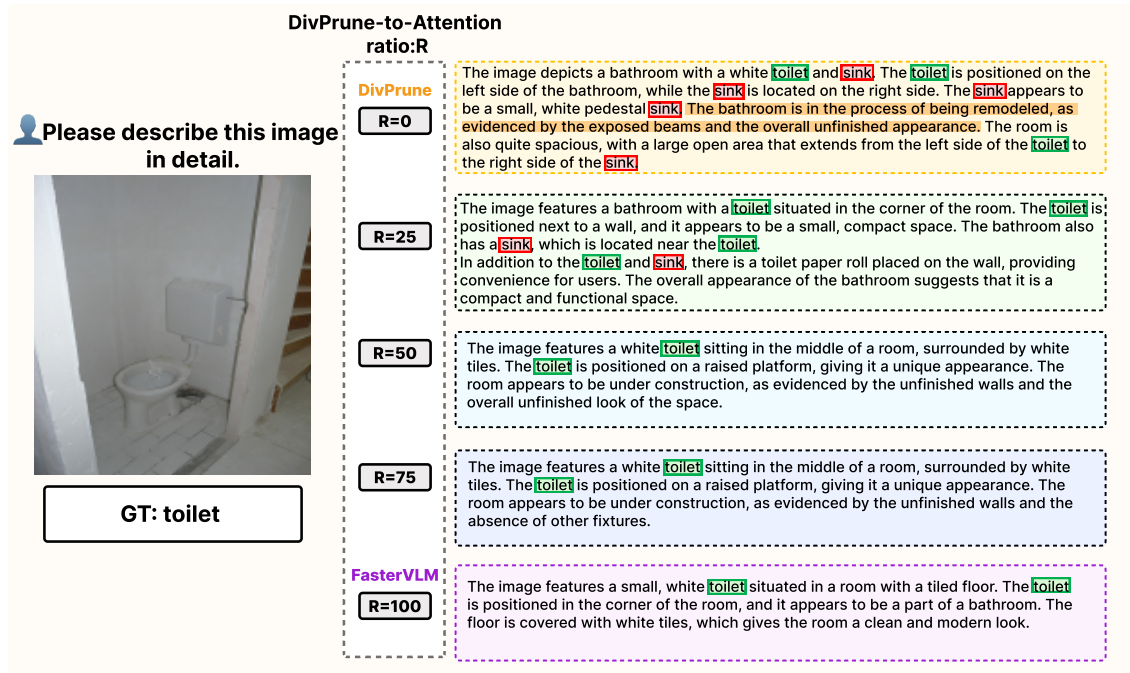


Figure 8: Effect of varying the DivPrune-to-Attention ratio  $R$  under a 64-token budget. As  $R$  decreases, hallucinated objects produced by DivPrune are progressively suppressed, and the responses shift from exploratory to fact-oriented descriptions. In the annotations, GT obj. and Hallucinated obj. label object words; denotes DivPrune-specific phrasing; red text indicates incorrect phrases.

## STATEMENT ON THE USE OF LARGE LANGUAGE MODELS

In the interest of transparency and in compliance with the ICLR 2026 guidelines, we report that a large language model (LLM) was used to assist in the refinement of this paper’s text.

**Scope of Use.** The model’s role was strictly limited to that of a writing assistant. Its contributions include:

- Correcting grammatical errors, spelling, and punctuation.
- Improving sentence structure and flow for enhanced clarity.
- Refining word choices for greater precision and conciseness.