Vision Token Pruning via Query–Vision Interaction Decomposition

Harshithanjani Athi [1] , **Sravan Kumar Ankireddy** [1] , **Jianzhong Charlie Zhang** [2] , **Hyeji Kim** [1]

Abstract

Large vision-language models (VLMs), which process both visual inputs and text queries, incur high inference costs due to the large number of visual tokens, making visual token pruning a natural approach for improving efficiency. Recent work has leveraged query information to guide token selection, often through heuristics derived from attention patterns or token dynamics. However, these approaches do not explicitly exploit the underlying structure of query–vision interactions. We address this gap by observing that the query–vision interaction matrix, naturally induced by the query and key projections already present in transformer attention, is effectively low-rank, with a small number of dominant latent interaction modes capturing query-relevant semantics. Motivated by this observation, we propose Query-Vision Decomposition (QViD), a novel training-free query-aware visual token pruning method that exploits this structure. Extensive experiments on both image and video understanding benchmarks show that QViD consistently improves performance under matched token budgets, with the largest gains appearing under aggressive compression regimes.

1. Introduction

Vision-language models (VLMs) have demonstrated strong performance across a wide range of multimodal tasks, including visual question answering, optical character recognition, diagram understanding, and video comprehension (Liu et al., 2024a; Wang et al., 2024; Li et al., 2024a). In a typical VLM, a vision encoder produces vision tokens from an image, and the language model jointly processes these vision tokens together with the text query to generate a final response. However, this design also creates a major efficiency bottleneck: an image can produce hundreds to thousands of vision tokens, making multimodal inference expensive in both latency and memory. Consequently, reducing the cost of processing vision tokens has become an important focus of recent work on efficient VLM inference.

A direct remedy is to reduce the number of visual tokens passed to the language model. Existing training-free methods either use attention-guided scores, as in FastV (Chen et al., 2024) and SparseVLM (Zhang et al., 2025), or attention-free token dynamics, as in V^{2}Drop (Chen et al., 2026). However, attention scores can suffer from positional bias, while variation-based scores capture generic token activity and only indirectly reflect the input query. Because visual relevance is query-dependent, an effective pruning rule should identify the image regions that support the current question rather than generic visual saliency, as illustrated in Figure 1. This motivates our central question:

> *Can a few dominant latent modes of query–vision interaction capture query-relevant visual evidence without relying on attention weights?*

Our Approach. To answer this question, we propose **Q**uery-**Vi**sion **D**ecomposition (QViD), a training-free visual-token pruning method based on an explicit query–vision interaction matrix. Our key observation is that this matrix has a strong low-rank structure: a small number of dominant interaction modes capture the visual evidence most relevant to the query. At an early transformer layer, QViD constructs this matrix and scores visual tokens by how strongly they participate in these dominant modes. Thus, unlike attention-guided pruning, QViD does not rely on attention weights, and unlike variation-based pruning, it explicitly conditions on the input query. As shown qualitatively in Figure 1, QViD focuses on image regions that provide direct evidence for the answer, while prior methods can highlight less relevant regions.

Our main contributions are:

- We show that query–vision interactions in VLMs, read-

[1]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA [2]Samsung Research America, Plano, TX, USA. Correspondence to: Harshithanjani Athi <ha26227.my@utexas.edu>, Hyeji Kim <hyeji.kim@utexas.edu>, Sravan Kumar Ankireddy <sravan.ankireddy@utexas.edu>, Jianzhong Charlie Zhang <jianzhong.z@samsung.com>.

ICML 2026 Workshop on Adaptation and Generalization in Foundation Models, Seoul, South Korea, 2026.

Figure 1. **Comparison of Visual-Token Scoring.** Compared with V^{2}Drop and SparseVLM, QViD concentrates high scores on query-relevant image regions, leading to more effective visual-token pruning.

ily obtainable from standard transformer query–key projections, exhibit a strong low-rank structure, where task-relevant semantics concentrate into a small number of dominant interaction modes.

- We propose QViD, a training-free SVD-based pruning method that scores visual tokens by their participation in these dominant interaction modes.

- Experiments across multiple VLMs and image and video understanding benchmarks show that QViD consistently outperforms strong recent pruning methods across a wide range of token budgets, with particularly strong gains under aggressive compression.

2. Related Work

Modern VLMs jointly process visual and textual tokens, and higher-resolution images or longer videos can produce hundreds to thousands of visual tokens (Liu et al., 2023; 2024a; Wang et al., 2024; Li et al., 2024a; Lin et al., 2023). This has motivated training-free visual-token compression for efficient VLM inference. Attention-guided methods such as FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2025), and HiRED (Arif et al., 2025) use LLM attention patterns, while PDrop (Xing et al., 2025) progressively removes visual redundancy. Token-merging methods such as ToMe (Bolya et al., 2023) and LLaVA-PruMerge (Shang et al., 2025) instead reduce sequence length by combining similar tokens.

A closely related attention-free method is V^{2}Drop (Chen et al., 2026), which prunes low-variation tokens across consecutive LLM layers. While this avoids explicit attention maps, its score reflects generic token activity rather than explicit query–vision relevance. In contrast, QViD is both attention-free and query-conditioned: it ranks visual tokens by their participation in the dominant low-rank structure of a query–vision interaction matrix.

3. Method

3.1. Problem Setup

We consider a vision-language model (VLM) that processes an input image and a text query jointly. The image is first encoded into a sequence of visual tokens, which are then combined with text tokens and processed by a transformer-based language model. At transformer layer ℓ, let

$$H_{\text{vis}}^{(\ell)} \in \mathbb{R}^{N_v \times d}, \qquad H_{\text{txt}}^{(\ell)} \in \mathbb{R}^{N_q \times d}$$

denote the hidden states corresponding to the visual tokens and user-query tokens, respectively, where N_v and N_q are the numbers of visual and query tokens, and d is the hidden dimension. Our goal is to prune *visual tokens* in a query-aware and training-free manner by retaining those most relevant to the input query under a target token budget. Our method can be applied at any layer ℓ, and we study this design choice empirically.

3.2. Query–Vision Interaction Matrix

Our method is based on a query–vision interaction matrix that measures the compatibility between query tokens and visual tokens. We construct this matrix using the frozen query and key projections already present in the transformer. For a given attention head at layer ℓ, we compute

$$Q_{\text{txt}} = H_{\text{txt}}^{(\ell)} W_Q, \qquad K_{\text{vis}} = H_{\text{vis}}^{(\ell)} W_K, \tag{1}$$

where $W_Q, W_K \in \mathbb{R}^{d \times d_h}$ are the query and key projection matrices, and d_h is the attention-head dimension. We then define the query–vision interaction matrix

$$M = Q_{\text{txt}} K_{\text{vis}}^{\top} \in \mathbb{R}^{N_q \times N_v}. \tag{2}$$

The entry M_{ij} measures the compatibility between query token i and visual token j in the model's learned representation space. Equivalently, the j-th column of M summarizes how visual token j interacts with the full query.

Advantages over Attention Scores. This construction is closely related to the pre-softmax attention computation, but differs in two important ways. First, we compute the

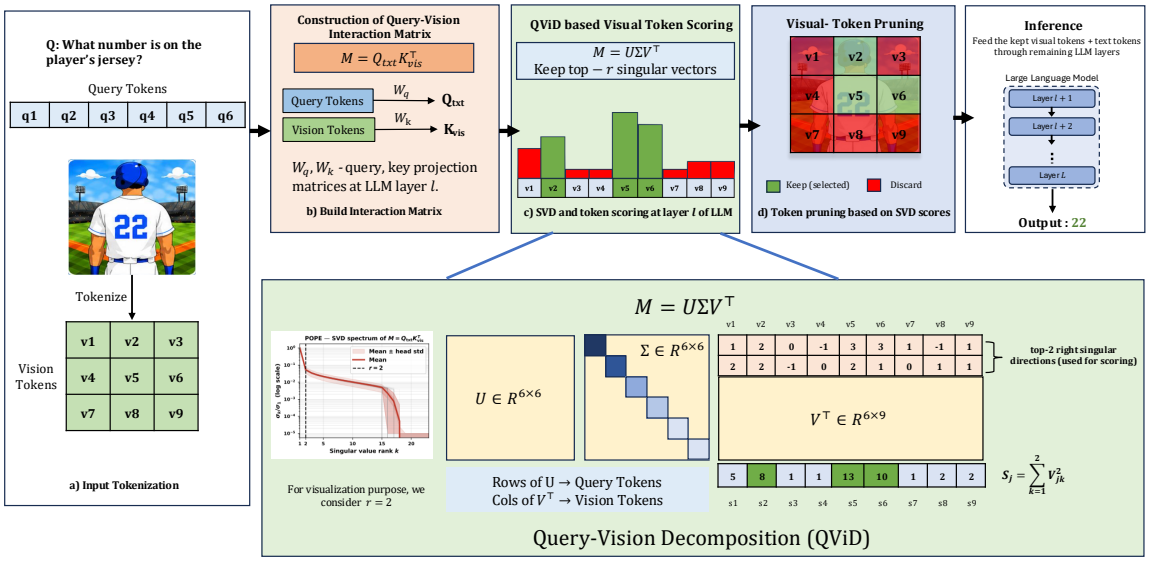

Figure 2. **QViD Pipeline.** QViD performs query-aware visual-token pruning by explicitly modeling query–vision interactions. At pruning layer ℓ, it forms the interaction matrix $M = Q_{\text{txt}} K_{\text{vis}}^{\top}$, extracts its dominant low-rank modes with SVD, and scores each visual token by its participation in these modes. The top-K tokens are retained for subsequent reasoning, preserving query-relevant visual evidence while reducing sequence length. Integer values in the illustrative matrices are used only for ease of presentation.

interaction before positional encoding. In autoregressive LLMs, positional encoding can bias attention toward visual tokens that are closer to the query tokens in the sequence, regardless of their semantic relevance. Since several prior pruning methods, such as SparseVLM and FastV, rely on attention scores for token selection, this positional bias can directly affect pruning decisions. By removing positional effects, M captures a purely content-based query–vision interaction signal.

Second, we avoid the normalization effects of softmax, where token relevance is mixed with competition among tokens. As a result, M preserves the raw query–vision interaction structure without relying on explicit attention maps. This is also advantageous in memory-efficient attention implementations, where attention maps may be unavailable or costly to extract.

From Interaction Matrix to Token Scores. A natural way to derive visual-token importance from M would be to aggregate each column independently, for example by taking its norm. This measures how strongly a visual token is compatible with all query tokens. However, as we show later in the ablation study, naive column-wise aggregation performs poorly as a pruning criterion.

This is because not every query token, or word, is equally informative or relevant for the downstream task. For example, in the question "What animal is wearing the hat?", we would like to emphasize compatibility with meaningful words such as *animal* and *hat*, rather than less informative words such as *is* or *the*.

More broadly, our goal is to capture visual evidence relevant to the *overall query semantics*, rather than simply measuring isolated token-level compatibility. As such relevance may not be recoverable from pairwise query–vision interactions alone, we instead analyze the global structure of the interaction matrix M itself.

3.3. Low-Rank Structure of Query–Vision Interactions and Dominant Interaction Modes

Interestingly, we observe that M, a quantity closely related to attention computation but not typically analyzed directly in prior pruning methods, exhibits a strong low-rank structure across a wide range of queries and images. Moreover, the dominant singular modes are not merely numerically significant; they consistently capture task-relevant query–vision interaction patterns.

Observation 1: Query–Vision Interactions Are Approximately Low-Rank. Across images and queries, the singular spectrum of M is strongly concentrated in a small number of directions. For each attention head, we compute the singular value decomposition (SVD) $M = U\Sigma V^{\top}$.

Figure 3 (a) shows the singular value spectrum of M for

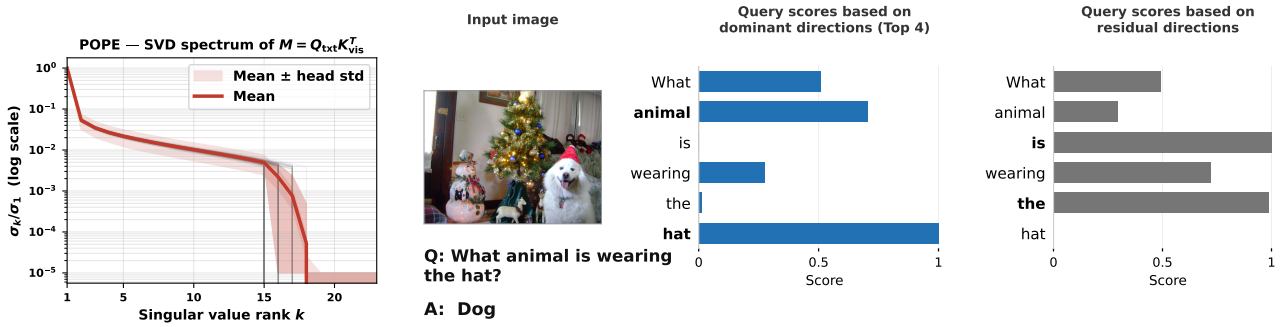

Figure 3. **Qualitative Observations. (a)** The query–vision interaction matrix has a rapidly decaying singular-value spectrum. **(b)** Dominant SVD components emphasize query-relevant tokens, while residual components are less aligned with the task.

LLaVA-1.5-7B on the POPE dataset (Li et al., 2023). We observe that the spectrum decays rapidly, with a small number of singular values explaining most of the query–vision interaction structure. As further shown in Appendix K.1, this behavior appears consistently across different datasets, images, and queries, suggesting that the useful interaction between query and visual tokens often lies in a low-dimensional subspace.

Observation 2: Dominant Components Capture Task-Relevant Structure. The low-rank structure of M is not only a numerical property; it is also semantically meaningful. In particular, we observe that task-relevant semantics tend to concentrate into a small number of dominant interaction modes.

For interpretability, we separate the leading singular directions, which capture the dominant interaction modes, from the remaining residual directions. We then measure how strongly each query token j participates in the dominant and residual subspaces through

$$s_j^{(\mathrm{dom})} = \sum_{k=1}^{4} U_{jk}^2, \qquad s_j^{(\mathrm{res})} = \sum_{k>4} U_{jk}^2. \tag{3}$$

Figure 3 (b) visualizes these two scores for the dog query–image example; additional qualitative examples are provided in Appendix K.2. We observe that semantically informative query tokens, such as *animal* and *hat*, tend to have high dominant contribution scores and low residual contribution scores. In contrast, less informative words, such as *is* and *the*, exhibit the opposite behavior.

Together, these observations suggest the following:

> *Visual-token importance should be estimated from the dominant interaction subspace of M, rather than from individual matrix entries or raw column magnitudes.*

3.4. QViD Token Scoring and Pruning

We now describe how QViD converts the query–vision interaction matrix into visual-token scores.

Visual-Token Scores. For each attention head at layer ℓ, we compute the singular value decomposition

$$M = U\Sigma V^\top. \tag{4}$$

The right singular vectors in V describe the visual-token side of the query–vision interaction modes. We score each visual token j by its participation in the top r right singular directions:

$$s_j = \sum_{k=1}^{r} V_{jk}^2. \tag{5}$$

Intuitively, V_{jk}^2 measures how strongly visual token j participates in the k-th dominant interaction mode. Therefore, a larger s_j indicates that the visual token j contributes more strongly to the dominant interaction subspace. For the multi-head setting, we apply the same scoring rule independently to each attention head and average the resulting scores. Let $M^{(h)}$ denote the query–vision interaction matrix for head h, and let $V^{(h)}$ be its right singular vector matrix. The final score for visual token j is

$$s_j = \frac{1}{H}\sum_{h=1}^{H}\sum_{k=1}^{r}\left(V_{jk}^{(h)}\right)^2, \tag{6}$$

where H is the number of attention heads.

For short queries, the number of available nonzero singular directions may be smaller than the chosen r. In this case, we use $r_{\mathrm{eff}} = \min(r, N_q, N_v)$, where N_q and N_v denote the numbers of query and visual tokens, respectively, and replace r by r_{eff} in the scoring rule.

Visual-Token Pruning. The top-K visual tokens with the largest scores s_j are passed to the remaining LLM layers, with K determined by the token budget. Importantly, QViD

Table 1. **Image Understanding Results on LLaVA-1.5-7B.** QViD achieves the best average performance at all retention ratios, with the largest gain under aggressive compression at 64 tokens. *Methods are sorted by average performance within each retention setting.*

Methods	GQA	SQA	TextVQA	POPE	MME	MMB	Average
Upper Bound, 576 Tokens							
LLaVA-1.5-7B (Liu et al. 2024)	61.9	69.5	58.2	85.9	1862	64.6	100.0%
Average Retain 192 Tokens (↓66.7%)							
QViD (ours)	**59.3**	69.0	**57.8**	**87.4**	1793	63.2	**98.1%**
V^2Drop [CVPR'26]	58.5	**69.3**	55.6	85.1	**1826**	**63.7**	97.6%
PDrop [CVPR'25]	57.1	68.8	56.1	82.3	1766	63.2	96.0%
SparseVLM [ICML'25]	57.6	69.1	56.1	83.6	1721	62.5	95.9%
HiRED [AAAI'25]	58.7	68.4	47.4	82.8	1737	62.8	93.6%
LLaVA-PruMerge [ICCV'25]	54.3	67.9	54.3	71.3	1632	59.6	90.3%
ToMe [ICLR'23]	54.3	65.2	52.1	72.4	1563	60.5	88.8%
FastV [ECCV'24]	52.7	67.3	52.5	64.8	1612	61.2	88.2%
Average Retain 128 Tokens (↓77.8%)							
QViD (ours)	**57.7**	68.7	**57.1**	**85.5**	**1726**	**61.9**	**96.4%**
V^2Drop [CVPR'26]	56.3	**68.8**	53.8	80.9	1712	61.8	94.0%
PDrop [CVPR'25]	56.0	68.3	54.8	82.3	1644	61.1	93.6%
SparseVLM [ICML'25]	56.0	67.1	54.9	80.5	1696	60.0	93.2%
HiRED [AAAI'25]	57.2	68.1	46.1	79.8	1710	61.5	91.6%
LLaVA-PruMerge [ICCV'25]	53.3	67.1	54.3	67.2	1554	58.1	87.9%
FastV [ECCV'24]	49.6	60.2	50.6	59.6	1490	56.1	81.7%
ToMe [ICLR'23]	52.4	59.6	49.1	62.8	1343	53.3	80.4%
Average Retain 64 Tokens (↓88.9%)							
QViD (ours)	52.2	68.3	**55.5**	**80.8**	**1553**	**56.3**	**90.4%**
V^2Drop [CVPR'26]	50.5	**68.9**	51.8	75.1	1470	55.2	86.9%
SparseVLM [ICML'25]	**52.7**	62.2	51.8	75.1	1505	56.2	86.5%
LLaVA-PruMerge [ICCV'25]	51.9	68.1	54.0	65.3	1549	55.2	86.5%
PDrop [CVPR'25]	41.9	68.6	45.9	55.9	1092	33.3	70.1%
FastV [ECCV'24]	46.1	51.1	47.8	48.0	1256	48.0	71.3%
ToMe [ICLR'23]	48.6	50.0	45.3	52.5	1138	43.7	69.7%

is entirely training-free, requires no architectural modification, and operates solely using the frozen projections already present in the model.

4. Experiments

4.1. Experimental Setup

We evaluate QViD on image and video understanding tasks using four representative VLMs: LLaVA-1.5-7B (Liu et al., 2024a), Qwen2-VL-7B (Wang et al., 2024), Video-LLaVA-7B (Lin et al., 2023), and LLaVA-OneVision-7B (Li et al., 2024a). For image understanding, we report results on GQA (Hudson & Manning, 2019), ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), POPE (Li et al., 2023), MME (Fu et al., 2023), and MMBench (Liu et al., 2024b); for video understanding, we use VideoMME (Fu et al., 2024) and MVBench (Li et al., 2024b). V^2Drop (Chen et al., 2026) is our primary baseline, and on LLaVA-1.5-7B we additionally compare against ToMe (Bolya et al., 2023), FastV (Chen et al., 2024), HiRED (Arif et al., 2025), LLaVA-PruMerge (Shang et al., 2025), SparseVLM (Zhang et al., 2025), and PDrop (Xing et al., 2025) using reported numbers from V^2Drop.

Unless otherwise stated, QViD prunes at LLM layer $\ell = 3$ using $r = 16$ SVD components and omits RoPE when constructing the query–vision interaction matrix. Additional multimodal reasoning, hallucination-sensitive, captioning, and implementation details are provided in the appendix.

Implementation Details. All experiments are conducted using Python 3.11, PyTorch 2.5.1 (CUDA 12.1). For LLaVA-1.5-7B, Qwen2-VL-7B, Video-LLaVA-7B, and LLaVA-OV-7B, we follow the standard evaluation settings of the corresponding model and benchmark. QViD is implemented as a separate scoring module at the pruning layer and does not modify the model architecture or attention kernel. After token selection, all subsequent transformer layers process the retained sequence normally. We use layer $\ell = 3$, $r = 16$ SVD components, and omit Rotary Positional Embedding (RoPE) when constructing the query–vision interaction matrix.

4.2. QViD: Main Results

Across all evaluated VLMs and benchmarks, QViD consistently improves performance under matched token budgets, with the largest gains under aggressive compression.

Table 2. **Generalization to Qwen2-VL and Video Understanding.** (a) Qwen2-VL-7B image understanding results across benchmarks. (b) LLaVA-OV-7B video understanding results.

(a) Qwen2-VL-7B Image Understanding

Method	AI2D	GQA	TextVQA	POPE	MME-P	MME-C	SQA	MMB	Avg.
Baseline	80.7	61.9	81.5	88.2	1684.0	635.7	85.3	84.9	100.0%
Retain 33.3%									
QViD	76.6	**58.6**	**79.0**	**85.6**	**1610.6**	493.6	**81.2**	81.0	**93.4%**
V^2Drop	**76.7**	52.8	53.4	85.4	1589.0	**600.4**	80.9	**82.0**	90.4%
Retain 22.2%									
QViD	**73.8**	**56.0**	**77.1**	**83.7**	**1577.9**	446.1	**79.0**	**78.8**	**90.1%**
V^2Drop	71.9	49.6	52.1	81.2	1563.3	**482.9**	78.4	78.6	84.8%
Retain 11.1%									
QViD	**69.8**	**50.9**	**69.1**	**78.4**	**1420.1**	**352.5**	**75.3**	**70.7**	**81.7%**
V^2Drop	65.0	39.7	45.0	57.6	863.0	294.6	70.1	48.4	62.7%

(b) LLaVA-OV-7B Video

Method	MVBench	Video MME	Avg.
Baseline	58.4	58.2	100.0%
Retain 25%			
QViD	**56.6**	**57.2**	**97.7%**
V^2Drop	56.5	57.1	97.5%
Retain 15%			
QViD	**54.0**	**55.7**	**94.2%**
V^2Drop	53.0	53.7	91.7%
Retain 10%			
QViD	**52.5**	**53.7**	**91.2%**
V^2Drop	39.4	41.4	69.4%

Compared with existing pruning methods, QViD more effectively preserves task-relevant visual information, leading to improved downstream accuracy on both image and video understanding tasks. This behavior is also reflected *qualitatively* in the retained visual tokens and score maps, as illustrated in Figure 1 and additional visualizations provided in Appendix M. On LLaVA-1.5-7B (Table 1), QViD achieves the best average performance at all three token budgets, improving over V^2Drop by 3.5% normalized average performance at 64 retained tokens, with strong gains on TextVQA, POPE, and MME. The gains also generalize to Qwen2-VL-7B (Table 2), where QViD improves average normalized performance across all retention ratios and gives a large improvement at 11.1% retention, from 62.7% to 81.7%. For video understanding, QViD outperforms V^2Drop on LLaVA-OV-7B at all token budgets (Table 2), preserving 91.2% of full-token performance at 10% retention compared with 69.4% for V^2Drop.

Additional Results. Beyond image and video understanding, we also evaluate whether the selected tokens preserve broader multimodal capabilities. Appendix H reports additional results on multimodal reasoning benchmarks, while Appendix J evaluates hallucination-sensitive performance and captioning quality. These results show that QViD preserves strong performance beyond standard perception tasks, suggesting that its selected tokens retain the visual information needed for downstream reasoning and generation.

Ablation Summary. We provide supporting ablations for the key design choices in Appendix B. First, we compare against a column-norm baseline that uses the same query–vision interaction matrix $M = Q_{\text{txt}} K_{\text{vis}}^{\top}$ but scores each visual token by $s_j^{\text{col}} = \|M_{:,j}\|_2^2$. This baseline performs substantially worse than QViD across all token budgets, showing that raw query–vision interaction strength alone is insufficient and that isolating the dominant low-rank interaction modes is important.

Second, we vary the number of SVD components n_{svd}. Performance improves sharply from $n_{\text{svd}} = 1, 2$ to $n_{\text{svd}} = 4, 8$, and $n_{\text{svd}} = 16$ provides strong and robust performance across budgets. These results support our design choice of scoring visual tokens using a small set of dominant query–vision interaction modes.

Computational Complexity. QViD adds modest query-aware scoring overhead but remains substantially cheaper than the full-token baseline. On LLaVA-1.5-7B, QViD reduces total compute from 8.86 TFLOPs to 2.53, 3.42, and 4.32 TFLOPs at 64, 128, and 192 retained tokens, respectively. Detailed compute, latency, and throughput results are provided in Appendix G and Table 8.

5. Conclusion

We presented QViD, a training-free visual-token pruning method that explicitly conditions on the query without relying on attention weights. QViD constructs the query–vision interaction matrix induced by standard transformer query–key projections and scores visual tokens using its dominant low-rank subspace, retaining tokens that participate in the main query-relevant interaction modes. Across image and video benchmarks and multiple VLMs, QViD improves performance under matched token budgets, with the largest gains under aggressive compression. These results show that a few dominant SVD components are sufficient to capture much of the task-relevant query–vision structure, establishing low-rank query–vision decomposition as a simple, interpretable, and training-free basis for efficient VLM inference.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work was partially supported by Samsung Research America through the 6G@UT center within the Wireless Networking and Communications Group (WNCG) at the University of Texas at Austin, ARO Award W911NF2310062, ONR Award N000142412542, and NSF CAREER Award 2443857.

References

Arif, K. H. I., Yoon, J., Nikolopoulos, D. S., Vandierendonck, H., John, D., and Ji, B. HiRED: Attention-guided token dropping for efficient inference of high-resolution vision-language models in resource-constrained environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your ViT but faster. In *Proceedings of the International Conference on Learning Representations*, 2023.

Chen, J., Liu, X., Wen, Z., Wang, Y., Huang, S., and Chen, H. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026.

Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., and Chang, B. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proceedings of the European Conference on Computer Vision*, 2024.

Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

Hudson, D. A. and Manning, C. D. 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*, pp. 6700–6709, 2019.

Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., and Li, C. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., and Wen, J.-R. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., and Yuan, L. Video-LLaVA: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26286–26296, 2024a.

Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., Chen, K., and Lin, D. MMBench: Is your multi-modal model an all-around player? In *Proceedings of the European Conference on Computer Vision*, pp. 216–233, 2024b.

Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, pp. 2507–2521, 2022.

Shang, Y., Cai, M., Xu, B., Lee, Y. J., and Yan, Y. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., et al. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zhang, Y., Fan, C.-K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., and Zhang, S. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the International Conference on Machine Learning*, 2025.

A. Limitations and Future Directions

Our evaluation is limited to public VLMs and standard image/video benchmarks, and may not fully capture behavior on closed-source models, proprietary inference stacks, or specialized domains. As a training-free pruning method, QViD also inherits the biases and failure modes of the underlying VLM. Practical speedups may vary with hardware, batching, sequence length, and attention-kernel implementations. Future work could extend query–vision decomposition to adaptive token budgets, dynamic selection of the number of SVD components and tighter integration with hardware-aware inference systems. It would also be valuable to study whether the same low-rank interaction structure can guide token pruning during training, in longer video settings, or in broader multimodal architectures.

B. Ablation Studies

We conduct ablation studies to examine the main design choices in QViD. In particular, we study whether the SVD-based decomposition is necessary, how many dominant SVD components are sufficient for token scoring, and how the choice of pruning layer affects performance.

Dominant Low-Rank Structure Is Important. To isolate the effect of SVD-based decomposition and low-rank interaction modeling, we compare QViD with a simple column-norm baseline that uses the same query–vision interaction matrix $M = Q_{\text{txt}} K_{\text{vis}}^{\top}$, but scores each visual token by raw interaction magnitude: $s_j^{\text{col}} = \|M_{:,j}\|_2^2$. This baseline directly measures how strongly each visual token interacts with the query, but does not separate dominant query-conditioned structure from residual interaction energy. Table 3 (a) shows that simply using the query–vision interaction matrix is not sufficient. Column-norm scoring performs much worse than QViD across all token budgets, especially under aggressive compression. The SVD step is crucial as it identifies the low-rank interaction modes that provide a more reliable signal for visual-token selection.

Effect of the Number of SVD Components. We study how many dominant SVD components are needed for reliable token scoring. Table 3 (b) reports the component sweep at 128 retained tokens. We denote the number of SVD components used for scoring by n_{svd}. A small number of dominant components is sufficient to recover most of the benefit of QViD: performance improves sharply from $n_{\text{svd}} = 1, 2$ to $n_{\text{svd}} = 4, 8$, indicating that the most relevant query–vision structure is concentrated in a few dominant interaction modes. Using $n_{\text{svd}} = 16$ provides additional robustness and achieves the best average performance at this budget. The full sweep over all token budgets is provided in Appendix D.

C. Layer Selection Results

We provide additional ablations on the choice of pruning layer ℓ. For each model, we compute the QViD scores using intermediate representations from layer ℓ and evaluate the resulting compressed model. This study tests whether token selection is more effective when computed from the raw input representations or after a few early layers of cross-token

Table 3. **Ablations on QViD Scoring with LLaVA-1.5-7B.** (a) SVD scoring outperforms column-norm scoring from the same query–vision interaction matrix. (b) Using only a small number of dominant SVD components recovers most of the benefit of QViD, indicating that token relevance is concentrated in a few interaction modes.

(a) Effect of SVD-Based Scoring

Method	GQA	SQA	TextVQA	POPE	MME	Avg.
Baseline	61.9	69.5	58.2	85.9	1862.0	100.0%
Retain 64 Tokens (↓88.9%)						
QViD	**52.2**	**68.3**	**55.5**	**80.8**	**1553.0**	**91.1%**
Column norm	42.2	64.0	42.6	54.7	1009.0	70.3%
Retain 128 Tokens (↓77.8%)						
QViD	**57.7**	**68.7**	**57.1**	**85.5**	**1726.0**	**96.5%**
Column norm	45.3	63.7	43.4	59.3	1069.0	73.2%
Retain 192 Tokens (↓66.7%)						
QViD	**59.3**	**69.0**	**57.8**	**87.4**	**1793.0**	**98.5%**
Column norm	47.8	63.0	43.6	65.0	1190.0	76.5%

(b) Number of SVD Components at 128 Tokens

Method	GQA	SQA	TextVQA	POPE	MME	Avg.
Baseline	61.9	69.5	58.2	85.9	1862.0	100.0%
V^2Drop	56.3	**68.8**	53.8	80.9	1712.0	93.7%
$n_{\text{svd}} = 1$	45.6	63.4	43.5	35.5	1077.1	67.8%
$n_{\text{svd}} = 2$	50.3	68.3	49.6	61.7	1478.6	83.2%
$n_{\text{svd}} = 4$	56.7	68.5	56.0	80.1	1697.0	94.2%
$n_{\text{svd}} = 8$	57.3	68.6	57.0	82.6	1725.1	95.6%
$n_{\text{svd}} = 16$	**57.7**	68.7	**57.1**	**85.5**	**1726.0**	**96.5%**

Table 4. **Layer Selection Ablation on LLaVA-1.5-7B.** We vary the scoring layer ℓ while retaining 128 visual tokens. MME denotes the sum of MME-Perception and MME-Cognition. Avg. denotes the mean normalized performance relative to the full-token baseline across all reported metrics. **Bold** denotes the best value in each column.

Layer ℓ	AI2D	GQA	TextVQA	POPE	MME	SQA	Avg.
Baseline	55.2	61.9	58.2	85.9	1862.0	69.5	100.0%
1	53.7	56.7	56.7	85.1	1718.8	68.9	96.1%
2	54.0	**57.9**	56.8	**85.6**	1710.9	68.6	96.5%
3	**54.1**	57.7	57.1	85.5	1726.3	68.7	**96.7%**
4	53.5	57.2	57.2	85.4	1719.3	**69.0**	96.4%
5	53.7	57.0	**57.3**	85.4	**1729.6**	68.6	96.5%
6	53.4	56.6	57.1	85.4	1678.6	68.4	95.7%

processing.

Table 4 reports the layer sweep for LLaVA-1.5-7B at 128 retained tokens. Table 5 reports the full Qwen2-VL-7B layer sweep across all three retention ratios.

For LLaVA-1.5-7B, performance is relatively stable across early layers. Layer $\ell = 3$ achieves the best average performance, but the gap between layers 1–5 is small, suggesting that QViD is not highly sensitive to the exact early layer used for scoring. The mild drop at layer 6 suggests that pruning too late may begin to discard visual tokens after task-relevant information has already been mixed across tokens.

For Qwen2-VL-7B, the choice of layer has a more visible effect, especially at lower retention ratios. Layer $\ell = 3$ gives the best average performance at 22.2% and 11.1% retention, while remaining within 0.2 percentage points of the best average at 33.3% retention. This suggests that a few early layers help refine the query–vision compatibility scores before pruning, while delaying pruning too much can reduce robustness under more aggressive compression. Overall, these results support using $\ell = 3$ as the default pruning layer in our experiments.

D. Effect of the Number of SVD Components

Here, n_{svd} denotes the number of singular-vector components used to compute the visual-token importance scores. The results show that QViD can capture useful query–vision structure with only a small number of components, but becomes more robust as more dominant interaction modes are included. In particular, performance improves sharply from $n_{\text{svd}} = 1, 2$ to $n_{\text{svd}} = 4, 8$, showing that single-mode scoring is insufficient under compression. The default setting $n_{\text{svd}} = 16$ achieves the best average performance at 128 and 192 retained tokens and remains close to the best setting at 64 tokens. This justifies using multiple SVD components in the final scoring rule and shows that the dominant query–vision subspace provides a stronger pruning signal than lower-rank alternatives.

E. Scoring Rule Ablation

We ablate different ways of converting the query–vision interaction matrix into visual-token scores. All variants use the same interaction matrix $M = Q_{\text{txt}} K_{\text{vis}}^{\top}$, but differ in how token importance is computed from M.

Compared Scoring Rules. **Column Norm** directly scores each visual token by the ℓ_2-norm of its corresponding column in M, without using an SVD decomposition. This measures the raw interaction magnitude between the visual token and the query tokens. **Sigma-Weighted** computes an SVD of the global interaction matrix and scores tokens by weighting each right singular-vector component by the corresponding singular value. **Per-Head Max** computes SVD-based scores independently for each attention head, and then aggregates scores by taking the maximum across heads. The default QViD scoring uses per-head SVD scores with the fixed aggregation rule used in the main experiments.

Table 7 shows that raw interaction strength alone is not sufficient for reliable token selection. Column-norm and sigma-weighted scoring perform identically because they induce the same token ordering. Indeed, if $M = U\Sigma V^{\top}$, then

$$M^{\top} M = V \Sigma^2 V^{\top}.$$

Table 5. **Full Layer Sweep on Qwen2-VL-7B.** We vary the QViD scoring layer ℓ across three retention ratios. Avg. is computed as the mean normalized performance relative to the full-token baseline. **Bold** denotes the best value in each column within a retention block.

Layer ℓ	AI2D	GQA	TextVQA	POPE	MME-P	MME-C	SQA	MMB	Avg.
Baseline	80.7	61.9	81.5	88.2	1684.0	635.7	85.3	84.9	100.0%
Retain 33.3% Tokens ($\downarrow 66.7\%$)									
1	74.8	57.2	78.5	84.9	1604.6	477.9	81.4	79.1	92.1%
2	75.7	58.0	**79.4**	85.9	1618.4	470.7	**81.7**	80.8	92.9%
3	**76.6**	58.6	79.0	85.6	1610.6	493.6	81.2	**81.0**	93.4%
4	75.9	**59.1**	77.9	86.7	1596.0	502.5	81.5	80.5	93.4%
5	74.8	**59.1**	76.4	**87.1**	1608.1	459.6	81.3	78.6	92.0%
6	74.5	57.4	78.2	85.1	**1627.1**	**543.2**	80.9	80.2	**93.6%**
Retain 22.2% Tokens ($\downarrow 77.8\%$)									
1	71.3	54.0	76.7	81.1	1504.4	426.4	78.1	73.5	87.0%
2	72.4	55.1	76.5	83.3	**1583.4**	384.3	79.6	76.2	88.1%
3	**73.8**	56.0	**77.1**	83.7	1577.9	446.1	79.0	**78.8**	**90.1%**
4	72.8	56.6	74.5	85.0	1519.3	**447.9**	**79.9**	76.2	89.2%
5	71.9	**57.3**	71.7	**85.6**	1499.5	367.1	79.8	74.1	86.8%
6	71.7	54.8	76.1	82.0	1556.8	426.1	78.6	75.9	88.1%
Retain 11.1% Tokens ($\downarrow 88.9\%$)									
1	66.4	47.5	68.4	73.4	1287.9	324.6	75.1	62.1	76.9%
2	68.9	49.1	66.8	77.0	1340.0	343.6	74.8	66.4	79.2%
3	**69.8**	**50.9**	**69.1**	78.4	**1420.1**	**352.5**	75.3	**70.7**	**81.7%**
4	68.7	50.3	63.9	79.5	1343.5	322.5	**75.6**	66.1	79.0%
5	66.5	**50.9**	58.9	**80.2**	1309.8	348.9	74.0	62.3	77.6%
6	67.9	48.7	65.6	75.7	1299.2	327.9	74.3	65.5	77.8%

Therefore, for visual token j,

$$\|M_{:,j}\|_2^2 = (M^\top M)_{jj} = \sum_k \sigma_k^2 V_{jk}^2.$$

Thus, sigma-weighted scoring is exactly the squared column-norm score. Since squaring is monotone for nonnegative scores, both methods select the same top-k visual tokens and give identical evaluation results. Their poor performance shows that selecting tokens by raw interaction energy is much weaker than selecting tokens by their participation in the dominant query–vision subspace.

In contrast, per-head SVD-based scoring performs substantially better, indicating that the main benefit comes from preserving head-specific low-rank query–vision structure rather than simply selecting tokens with large interaction magnitude.

The per-head max variant is competitive with the default QViD aggregation and performs slightly better on some metrics, especially at smaller token budgets. This suggests that different attention heads may specialize in different query-relevant visual cues. In this work, we keep the default aggregation fixed for consistency across models and budgets. Adaptive or query-dependent head aggregation is an interesting direction for future work.

F. Efficient QViD Scoring

QViD scores visual tokens using the dominant right singular directions of the query–vision interaction matrix. For each attention head h, we form

$$M^{(h)} = Q_{\text{txt}}^{(h)} \left(K_{\text{vis}}^{(h)}\right)^\top \in \mathbb{R}^{N_q \times N_v},$$

where N_q is the number of query tokens and N_v is the number of visual tokens. A direct implementation could compute the SVD of $M^{(h)}$ and use its top right singular vectors for token scoring. However, since $N_q \ll N_v$, we can compute the same right-singular subspace more efficiently using a QR decomposition.

Method	GQA	SQA	TextVQA	POPE	MME	Avg.
Baseline	61.9	69.5	58.2	85.9	1862	100.0%
Retain 64 Tokens (↓88.9%)						
V^{2}Drop	50.5	**68.9**	51.8	75.1	1470.0	87.2%
$n_{\text{svd}} = 1$	42.1	63.8	42.6	20.7	1002.7	62.2%
$n_{\text{svd}} = 2$	47.4	67.5	48.3	54.8	1362.3	78.7%
$n_{\text{svd}} = 4$	53.5	68.1	54.2	74.0	1599.5	89.9%
$n_{\text{svd}} = 8$	**54.1**	68.2	**55.5**	76.0	**1623.8**	**91.3%**
$n_{\text{svd}} = 16$	52.2	68.3	**55.5**	**80.8**	1553.0	91.1%
Retain 128 Tokens (↓77.8%)						
V^{2}Drop	56.3	**68.8**	53.8	80.9	1712.0	93.7%
$n_{\text{svd}} = 1$	45.6	63.4	43.5	35.5	1077.1	67.8%
$n_{\text{svd}} = 2$	50.3	68.3	49.6	61.7	1478.6	83.2%
$n_{\text{svd}} = 4$	56.7	68.5	56.0	80.1	1697.0	94.2%
$n_{\text{svd}} = 8$	57.3	68.6	57.0	82.6	1725.1	95.6%
$n_{\text{svd}} = 16$	**57.7**	68.7	**57.1**	**85.5**	**1726.0**	**96.5%**
Retain 192 Tokens (↓66.7%)						
V^{2}Drop	58.5	**69.3**	55.6	85.1	**1826.0**	97.4%
$n_{\text{svd}} = 1$	48.4	63.2	43.8	50.3	1204.7	73.5%
$n_{\text{svd}} = 2$	55.1	68.5	51.5	73.6	1612.7	89.7%
$n_{\text{svd}} = 4$	58.2	68.9	57.1	83.1	1768.2	96.6%
$n_{\text{svd}} = 8$	59.1	68.3	57.6	84.8	1788.2	97.5%
$n_{\text{svd}} = 16$	**59.3**	69.0	**57.8**	**87.4**	1793.0	**98.5%**

Table 6. **Ablation on the Number of SVD Components for LLaVA-v1.5-7B.** n_{svd} denotes the number of singular-vector components used to compute the visual-token importance scores. We vary n_{svd}, the number of singular-vector components used for QViD scoring, across three token budgets. Avg. denotes the mean normalized performance relative to the full-token baseline across all reported metrics.

QR-Based Reduction. Consider the transpose

$$\left(M^{(h)}\right)^\top \in \mathbb{R}^{N_v \times N_q},$$

which is tall and thin. We first compute its thin QR decomposition:

$$\left(M^{(h)}\right)^\top = \widehat{Q}^{(h)} R^{(h)}, \qquad \widehat{Q}^{(h)} \in \mathbb{R}^{N_v \times N_q}, \quad R^{(h)} \in \mathbb{R}^{N_q \times N_q},$$

where the columns of $\widehat{Q}^{(h)}$ are orthonormal and $R^{(h)}$ is upper triangular. We then compute the SVD of the small matrix $\left(R^{(h)}\right)^\top$:

$$\left(R^{(h)}\right)^\top = \widetilde{U}^{(h)} \Sigma^{(h)} \left(\widetilde{V}^{(h)}\right)^\top$$

Substituting the QR factorization into $M^{(h)}$, we obtain

$$\begin{aligned} M^{(h)} &= \left(R^{(h)}\right)^\top \left(\widehat{Q}^{(h)}\right)^\top \\ &= \widetilde{U}^{(h)} \Sigma^{(h)} \left(\widetilde{V}^{(h)}\right)^\top \left(\widehat{Q}^{(h)}\right)^\top \\ &= \widetilde{U}^{(h)} \Sigma^{(h)} \left(\widehat{Q}^{(h)} \widetilde{V}^{(h)}\right)^\top \end{aligned}$$

Therefore, the right singular vectors of $M^{(h)}$ are

$$V^{(h)} = \widehat{Q}^{(h)} \widetilde{V}^{(h)} \in \mathbb{R}^{N_v \times N_q}$$

Thus, the QR-based computation is mathematically equivalent to computing the right singular vectors of $M^{(h)}$ directly, but avoids applying SVD to the full rectangular $N_q \times N_v$ matrix.

Table 7. **Scoring-Rule Ablation on LLaVA-1.5-7B.** We compare different scoring rules derived from the query–vision interaction matrix at matched token budgets. Avg. denotes the mean normalized performance relative to the full-token baseline across all reported metrics.

Method	GQA	SQA	TextVQA	POPE	MME	Avg.
Baseline	61.9	69.5	58.2	85.9	1862.0	100.0%
Retain 64 Tokens (↓88.9%)						
Column norm	42.2	64.0	42.6	54.7	1009.0	70.3%
Sigma-weighted	42.2	64.0	42.6	54.7	1009.0	70.3%
Per-head max	**53.2**	68.1	**55.7**	**82.1**	**1581.0**	**92.0%**
QViD	52.2	**68.3**	55.5	80.8	1553.0	91.1%
Retain 128 Tokens (↓77.8%)						
Column norm	45.3	63.7	43.4	59.3	1069.0	73.2%
Sigma-weighted	45.3	63.7	43.4	59.3	1070.0	73.2%
Per-head max	57.6	68.4	**57.1**	**86.3**	**1734.0**	**96.6%**
QViD	**57.7**	**68.7**	**57.1**	85.5	1726.0	96.5%
Retain 192 Tokens (↓66.7%)						
Column norm	47.8	63.0	43.6	65.0	1190.0	76.5%
Sigma-weighted	47.8	63.0	43.6	65.0	1190.0	76.5%
Per-head max	**59.4**	67.9	**58.0**	87.2	1780.0	98.1%
QViD	59.3	**69.0**	57.8	**87.4**	**1793.0**	**98.5%**

Token Scoring. Let $V^{(h)}_{r_{\text{eff}}} \in \mathbb{R}^{N_v \times r_{\text{eff}}}$ denote the first r_{eff} columns of $V^{(h)}$. In our experiments, we set $r = n_{\text{svd}}$ and use

$$r_{\text{eff}} = \min(n_{\text{svd}}, N_q, N_v),$$

so the score is well defined even for short queries. The per-head visual-token score is the row-leverage score

$$s_j^{(h)} = \sum_{k=1}^{r_{\text{eff}}} \left(V_{jk}^{(h)}\right)^2$$

We average these scores across attention heads:

$$s_j = \frac{1}{H} \sum_{h=1}^{H} s_j^{(h)}$$

The top-K visual tokens by s_j are retained and passed to the remaining LLM layers.

Complexity. For each head, the QR decomposition of $\left(M^{(h)}\right)^\top \in \mathbb{R}^{N_v \times N_q}$ costs $O(N_v N_q^2)$, and the SVD of the small $N_q \times N_q$ matrix costs $O(N_q^3)$. Across H heads, the scoring complexity is

$$O(H N_v N_q^2 + H N_q^3)$$

Since N_q is typically much smaller than N_v, this cost is small compared with the cost of forwarding all visual tokens through the LLM. In our LLaVA-1.5-7B experiments, the full scoring step, including construction of $M = Q_{\text{txt}} K_{\text{vis}}^\top$, costs less than 50 MFLOPs.

Implementation. The QR-based scoring step can be implemented compactly as follows:

```
# M: (H, N_q, N_v), n_svd = number of SVD components
r_eff = min(n_svd, M.shape[-2], M.shape[-1])

Q_qr, R_qr = torch.linalg.qr(M.mT)
_, _, Vh = torch.linalg.svd(R_qr.mT, full_matrices=False)
```

Table 8. **Compute Comparison on LLaVA-1.5-7B.** Total TFLOPs include pruning overhead. Reduction is computed relative to the full-token baseline.

Method	Tokens	TFLOPs	Reduction
Baseline	576	8.86	–
V^2Drop	64	1.97	77.7%
QViD	64	2.53	71.5%
V^2Drop	128	2.84	67.9%
QViD	128	3.42	61.4%
V^2Drop	192	3.72	57.9%
QViD	192	4.32	51.3%

```
# Right singular vectors of M
W = (Q_qr @ Vh.mT)[:, :, :r_eff]  # (H, N_v, r_eff)

# Visual-token leverage scores, averaged over heads
scores = (W * W).sum(dim=-1).mean(dim=0)  # (N_v,)
```

G. Detailed Efficiency Results

In this section, we provide detailed latency, throughput, and FLOPs measurements for LLaVA-1.5-7B. The main paper reports the high-level compute comparison. Here, we include the full efficiency breakdown used for the complexity analysis.

Scoring Overhead. QViD adds a lightweight query–vision scoring pass before pruning: with the QR-based implementation described in Appendix F, the scoring cost is $O(HN_vN_q^2 + HN_q^3)$ for H attention heads, N_q query tokens, and N_v visual tokens. Since $N_q \ll N_v$ in typical VLM inputs, this overhead is small relative to forwarding all visual tokens through the LLM and is amortized by pruning; detailed derivations and efficiency measurements are provided in Appendices F as well as Section 4.2.

Compute Accounting. For the full-token baseline, the model processes all 576 visual tokens. For V^2Drop, the reported TFLOPs include its per-layer token schedule, where early layers retain the full visual sequence before pruning. For QViD, the reported TFLOPs include the early query-aware scoring pass and the subsequent LLM computation using the retained visual tokens. The QR/SVD scoring computation itself is small, costing less than 50 MFLOPs including construction of the interaction matrix.

From our FLOP measurements on LLaVA-1.5-7B, removing one visual token from the remaining LLM computation saves roughly 0.01 TFLOPs. Therefore, retaining 64, 128, and 192 tokens instead of all 576 visual tokens saves approximately 7.17, 6.27, and 5.38 TFLOPs, respectively, in the remaining LLM computation. This shows that the QViD scoring overhead is much smaller than the compute saved by pruning.

Latency and Throughput. QViD has slightly higher latency than V^2Drop at the same retained-token budget because it computes query-aware scores before pruning. However, it still improves throughput over the full-token baseline. The full baseline runs at 8.60 images/s, while QViD runs at 10.40, 10.27, and 10.26 images/s for 64, 128, and 192 retained tokens, respectively. Thus, QViD introduces modest additional overhead relative to V^2Drop, but remains substantially more efficient than the full-token baseline.

Implementation details. All experiments are conducted using Python 3.11, PyTorch 2.5.1, and CUDA 12.1 on a single NVIDIA A100-80GB GPU. For LLaVA-1.5-7B, Qwen2-VL-7B, Video-LLaVA-7B, and LLaVA-OV-7B, we follow the standard evaluation settings of the corresponding model and benchmark. Unless otherwise stated, QViD prunes at LLM layer $\ell = 3$ using $n_{\text{svd}} = 16$ dominant SVD components, and omits Rotary Positional Embedding (RoPE) when constructing the query–vision interaction matrix. Reported TFLOPs include both the pruning overhead and the subsequent LLM computation on the retained visual tokens.

Table 9. **Detailed Efficiency Results on LLaVA-1.5-7B.** We report measured latency, throughput, speedup, and total TFLOPs for different pruning methods and retained-token budgets. TFLOPs include pruning overhead.

Method	Tokens	Lat. (ms)	Tput. (img/s)	Speedup	TFLOPs
Baseline	576	100.1	8.60	1.00×	8.86
QViD	64	80.1	10.40	1.21×	2.53
QViD	128	81.3	10.27	1.19×	3.42
QViD	192	81.5	10.26	1.19×	4.32
V^2Drop	64	77.7	10.66	1.24×	1.97
V^2Drop	128	78.6	10.55	1.23×	2.85
V^2Drop	192	78.2	10.58	1.23×	3.73

Method	n	AI2D	ChartQA	MMMU	OCRBench	SEEDBench	Avg.
Baseline	576	55.2	18.2	36.1	31.5	66.2	100.0%
Retain 64 Tokens (↓ 88.9%)							
V^2Drop	64	**52.3**	13.2	33.3	14.4	48.0	75.5%
QViD	64	52.2	**16.6**	**35.0**	**26.5**	**54.6**	**89.9%**
Retain 128 Tokens (↓ 77.8%)							
V^2Drop	128	53.5	15.0	**35.1**	26.2	59.4	89.9%
QViD	128	**54.1**	**17.0**	**35.1**	**30.5**	**61.0**	**95.5%**
Retain 192 Tokens (↓ 66.7%)							
V^2Drop	192	**54.7**	15.8	35.7	26.8	**63.8**	93.3%
QViD	192	54.3	**17.4**	**37.1**	**30.9**	63.5	**98.2%**

Table 10. **Additional Multimodal Benchmark Results on LLaVA-v1.5-7B.** We compare QViD with V^2Drop at matched token budgets on AI2D, ChartQA, MMMU, OCRBench, and SEEDBench. Avg. denotes the mean normalized performance relative to the full-token baseline.

H. Additional Multimodal Benchmark Results

We further evaluate QViD on additional multimodal benchmarks beyond the main evaluation suite. For LLaVA-v1.5-7B, we report results on AI2D, ChartQA, MMMU, OCRBench, and SEEDBench. For Qwen2-VL-7B, we additionally evaluate on MMMU-Pro Standard, a more challenging multimodal reasoning benchmark with 10 answer options.

Table 10 shows that QViD consistently improves over V^2Drop in average normalized performance across additional LLaVA-v1.5-7B benchmarks. The gains are especially large on OCRBench and ChartQA, suggesting that query-aware token selection preserves visually grounded information beyond the main evaluation suite.

Table 11 reports results on MMMU-Pro Standard for Qwen2-VL-7B. In the standard-format split, the model receives the question, answer choices, and associated image as structured multimodal input. QViD remains close to V^2Drop at 33.3% and 22.2% retention, while substantially improving performance at 11.1% retention, from 23.6% to 28.8%. This indicates that query-aware token selection is especially beneficial under aggressive compression on challenging multimodal reasoning tasks.

I. Additional Video Understanding Results

Table 12 reports additional video understanding results on Video-LLaVA-7B. The trends are consistent with the LLaVA-OV-7B results in the main paper: QViD outperforms V^2Drop at all retention ratios, with the largest gain under the most

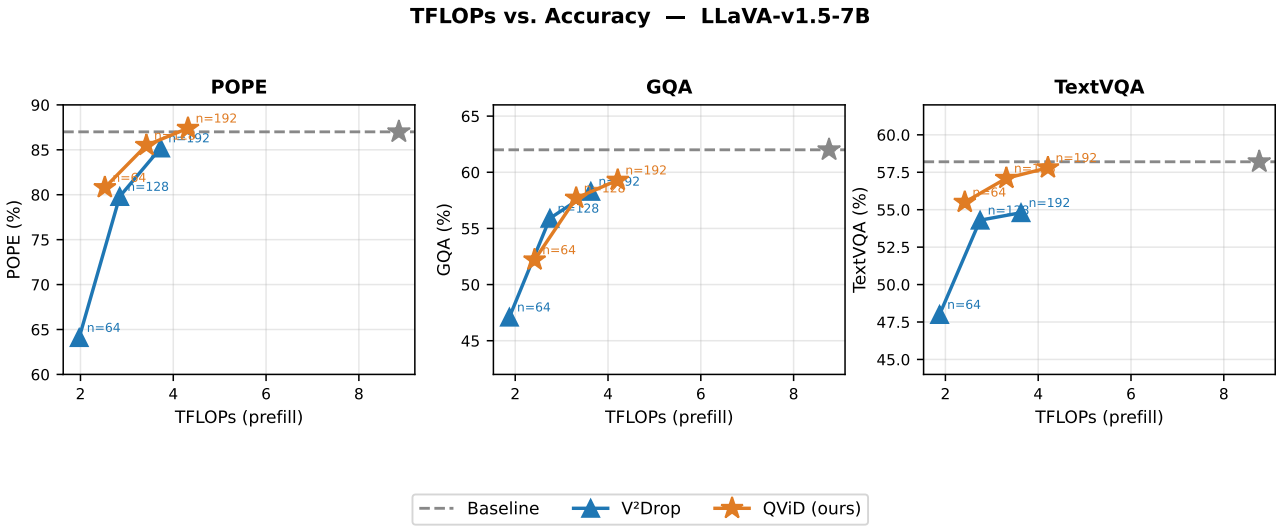

Figure 4. **Accuracy–Compute Tradeoff on LLaVA-1.5-7B.** We compare V^{2}Drop, and QViD at 64, 128, and 192 retained visual tokens on POPE, GQA, and TextVQA. The gray dashed line and star indicate the full-token baseline with 576 visual tokens. QViD achieves a stronger accuracy–compute tradeoff, especially under aggressive pruning, by retaining more query-relevant visual tokens at comparable TFLOPs.

Table 11. **MMMU-Pro Standard Results on Qwen2-VL-7B.** We compare QViD with V^{2}Drop under matched retention ratios on the 1,730-sample standard-format split.

Method	Retention	MMMU-Pro
Baseline	100%	33.8
V^{2}Drop	33.3%	**32.3**
QViD	33.3%	31.7
V^{2}Drop	22.2%	**30.2**
QViD	22.2%	29.9
V^{2}Drop	11.1%	23.6
QViD	11.1%	**28.8**

aggressive compression setting.

J. Hallucination and Captioning Results

We further evaluate QViD on HallusionBench and COCO Captioning to test whether token compression preserves both hallucination robustness and open-ended generation quality. HallusionBench is a hallucination-sensitive benchmark and is reported using all-pair accuracy (aAcc), figure accuracy (fAcc), and question accuracy (qAcc). COCO Captioning is evaluated using CIDEr, a caption-quality metric based on consensus between generated captions and human reference captions. CIDEr is not an accuracy metric; in our lmms-eval setup it is reported on a normalized scale, where values around 1.0 roughly correspond to a traditional CIDEr score around 100. Higher is better for all metrics.

Tables 13 and 14 show that QViD remains effective on hallucination-sensitive and generative benchmarks. On LLaVA-v1.5-7B, QViD improves over V^{2}Drop across all HallusionBench metrics and CIDEr at every token budget. On Qwen2-VL-7B, QViD also improves HallusionBench performance and gives substantially higher CIDEr at matched retention ratios. These results suggest that query-aware token selection preserves visual information needed for both grounded recognition and open-ended caption generation.

Table 12. **Video Understanding Results on Video-LLaVA-7B.** We compare QViD with V^{2}Drop on MVBench and VideoMME under token compression. Avg. denotes the mean normalized performance over MVBench and VideoMME Overall relative to the full-token baseline.

Method	MVBench	VideoMME				Avg.
		Overall	Short	Med.	Long	
All Tokens (100%)						
Baseline	42.9	36.2	41.2	35.0	32.4	100.0%
Retention = 25%						
QViD	**41.5**	**37.7**	**43.2**	**36.1**	**33.8**	**100.4%**
V^{2}Drop	41.2	36.3	40.1	35.1	33.7	98.2%
Retention = 15%						
QViD	**39.8**	**35.6**	**38.7**	**34.4**	**33.8**	**95.6%**
V^{2}Drop	38.4	32.8	35.3	32.3	30.8	90.1%
Retention = 10%						
QViD	**39.5**	**34.7**	**36.9**	**34.1**	**33.2**	**94.0%**
V^{2}Drop	33.6	29.9	29.4	29.8	30.4	80.5%

Table 13. **HallusionBench and COCO Captioning Results on LLaVA-v1.5-7B.** We compare QViD with V^{2}Drop at matched token budgets. HallusionBench is reported using aAcc, fAcc, and qAcc, while COCO Captioning is reported using CIDEr. QViD improves both hallucination-related metrics and caption quality under visual-token compression.

Method	n	HallusionBench			COCO Captioning
		aAcc	fAcc	qAcc	CIDEr
Baseline	576	47.9	20.5	12.3	1.087
Retain 64 Tokens ($\downarrow$ 88.9%)					
V^{2}Drop	64	46.8	15.0	9.5	0.591
QViD	64	**47.8**	**19.7**	**12.7**	**0.782**
Retain 128 Tokens ($\downarrow$ 77.8%)					
V^{2}Drop	128	47.1	17.6	11.9	0.939
QViD	128	**48.4**	**20.8**	**12.7**	**0.967**
Retain 192 Tokens ($\downarrow$ 66.7%)					
V^{2}Drop	192	47.3	18.2	11.6	0.995
QViD	192	**48.6**	**20.8**	**12.7**	**1.028**

K. Additional Qualitative Analysis

K.1. Low-Rank Structure of Query–Vision Interactions

A key motivation for QViD is that the query–vision interaction matrix M has a concentrated singular spectrum. To verify this, we compute the singular values of $M = Q_{\text{txt}} K_{\text{vis}}^{\top}$ at the pruning layer across multiple evaluation datasets. Figure 5 plots the normalized spectrum σ_k/σ_1 on a logarithmic scale.

Across all datasets, the spectrum decays rapidly after the first few components, showing that most of the query–vision interaction energy is concentrated in a small number of dominant directions. This supports our design choice of using only the top SVD components for token scoring. The trend is consistent across POPE, GQA, TextVQA, MMBench, ScienceQA, and MME, suggesting that the low-rank structure is not specific to a single benchmark or task type. Although the exact decay rate varies across datasets, the dominant components consistently capture the strongest query-conditioned interaction modes.

K.2. Qualitative Analysis of SVD Components

For the query-token visualizations, we use the analogous left-singular-vector score. If $M = U\Sigma V^{\top}$ and U_r contains the top r left singular vectors, the query-token score is $q_i = \sum_{k=1}^{r} U_{ik}^2$. Figure 6 compares scores from the dominant SVD

Table 14. **HallusionBench and COCO Captioning Results on Qwen2-VL-7B.** We compare QViD with V^2Drop at matched retention ratios where available. HallusionBench is reported using aAcc, fAcc, and qAcc, while COCO Captioning is reported using CIDEr. QViD preserves stronger hallucination robustness and substantially better captioning quality under compression.

Method	Retention	HallusionBench			COCO Captioning
		aAcc	fAcc	qAcc	CIDEr
Baseline	100%	67.5	38.7	42.2	0.977
Retain 33.3% Tokens (↓ 66.7%)					
V^2Drop	33.3%	61.8	33.8	33.6	0.563
QViD	33.3%	**62.3**	**35.3**	**35.2**	**0.922**
Retain 22.2% Tokens (↓ 77.8%)					
V^2Drop	22.2%	57.0	27.7	25.9	0.554
QViD	22.2%	**60.3**	**31.2**	**31.2**	**0.880**
Retain 11.1% Tokens (↓ 88.9%)					
QViD	11.1%	58.1	28.0	27.3	0.752

Table 15. **Confidence Intervals for LLaVA-1.5-7B Results.** We report mean ± standard error for QViD and V^2Drop at different retained-token budgets. Mean values are aligned with Table 1.

Method	Tokens	POPE	TextVQA	GQA	MME
QViD	64	80.8 ± 0.4141	55.5 ± 0.6737	52.2 ± 0.4441	1553 ± 0.0318
V^2Drop	64	75.1 ± 0.5056	51.8 ± 0.6775	50.5 ± 0.4451	1470 ± 0.0323
QViD	128	85.5 ± 0.3764	57.1 ± 0.6715	57.7 ± 0.4404	1726 ± 0.0315
V^2Drop	128	80.9 ± 0.4232	53.8 ± 0.6736	56.3 ± 0.4427	1712 ± 0.0320
QViD	192	87.4 ± 0.3572	57.8 ± 0.6695	59.3 ± 0.4384	1793 ± 0.0316
V^2Drop	192	85.1 ± 0.3744	55.6 ± 0.6731	58.5 ± 0.4396	1826 ± 0.0313

components with scores from the residual components obtained after removing the top directions. The dominant components assign high scores to task-critical words, such as *country*, *coin*, *animal*, and *hat*, which are directly tied to the answer. In contrast, the residual components place more weight on less informative or function words, indicating weaker alignment with the task. The singular-value spectra in the same figure also show rapid decay, suggesting that the main query–vision interaction structure is concentrated in a few dominant directions. This supports our choice of using only the dominant SVD components for visual-token scoring.

L. Confidence Intervals

To quantify the statistical reliability of the reported results, we provide confidence intervals for the main LLaVA-1.5-7B comparison in Table 15. For each benchmark, we report the mean performance together with the standard error. The mean values are matched to the corresponding entries in Table 1. Across token budgets, QViD maintains consistent gains over V^2Drop on POPE, TextVQA, and GQA, with especially large margins at 64 retained tokens. These results indicate that the improvements are not limited to a single metric or token budget, and remain stable under aggressive compression.

M. Additional Token-Score Visualizations

Figure 7 shows additional token-score maps comparing QViD with V^2Drop and SparseVLM. Across diverse image-query pairs, QViD assigns high scores to localized regions that are relevant to the question, such as the snowboarder, bat, scoreboard, license plate, and player jersey. In contrast, the baseline score maps are often more diffuse or place high scores on less query-relevant regions. These examples provide qualitative evidence that query–vision SVD scoring better aligns token importance with the visual evidence needed to answer the input query.

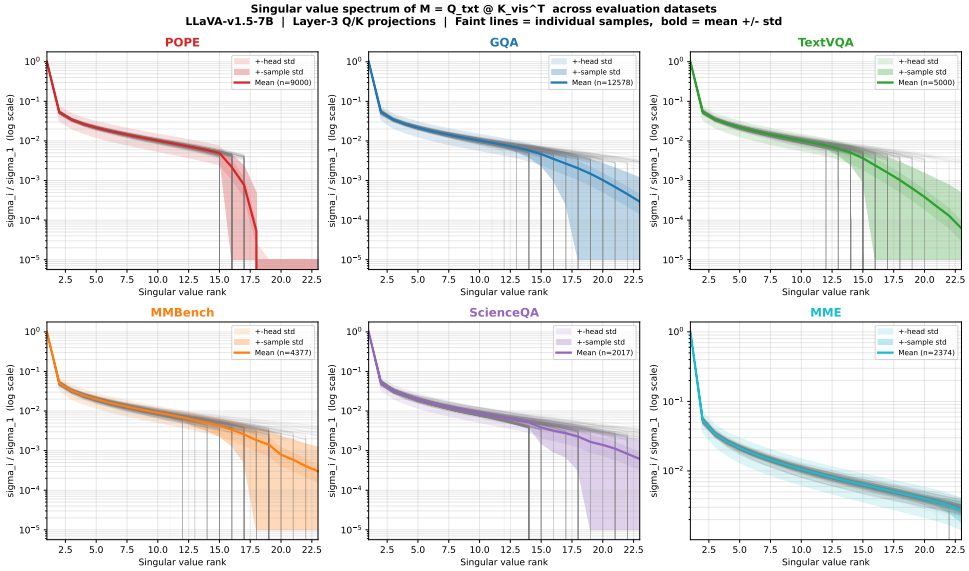

Figure 5. **Low-Rank Structure of Query–Vision Interactions.** We plot the normalized singular-value spectrum σ_k/σ_1 of $M = Q_{\text{txt}} K_{\text{vis}}^{\top}$ at the pruning layer of LLaVA-1.5-7B. Across datasets, the spectra decay rapidly, showing that query–vision interactions are concentrated in a few dominant directions. Faint curves show individual samples, solid curves show the mean, and shaded bands indicate variation across heads.

M.1. Query-Token Scores from Dominant and Residual Components

Figure 8 provides additional examples comparing query-token scores from the dominant SVD components and the residual components. For each image-query pair, we compute scores using the top SVD components and compare them with scores from the residual directions. The dominant components tend to emphasize task-relevant query words, such as *country*, *state*, *lions*, *score*, *batter*, *liquor*, and *jersey*. In contrast, the residual components often assign larger scores to function words or generic question tokens. This supports our observation that the dominant query–vision interaction modes capture semantic structure useful for answering the query.

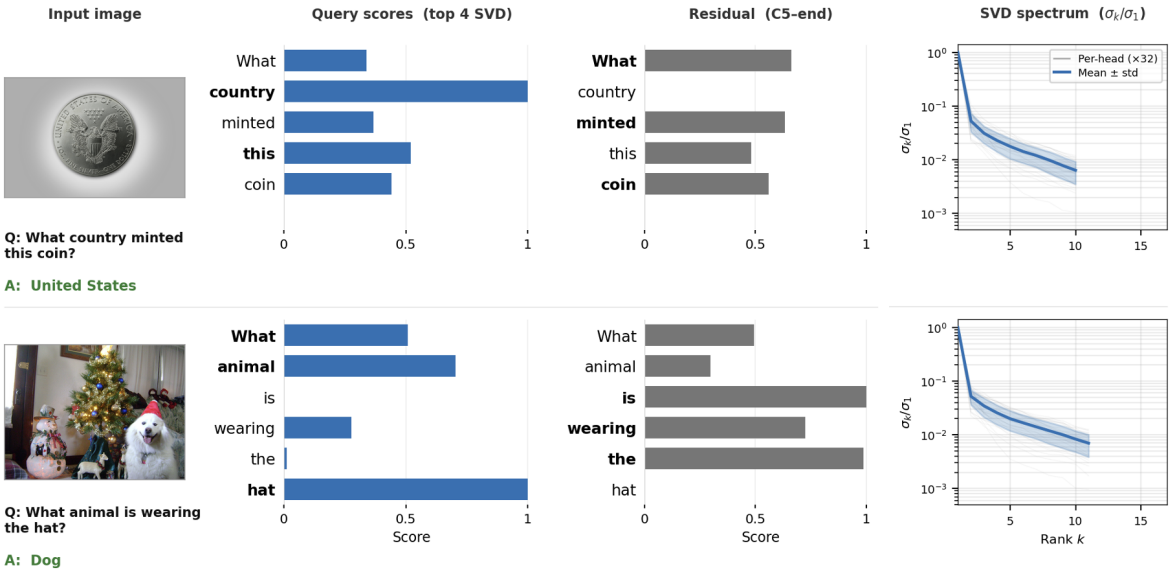

Figure 6. **Dominant SVD Components Capture Task-Relevant Structure.** Top SVD components emphasize task-critical query tokens, while residual components are less aligned with the question. The spectra show that query–vision interactions are concentrated in a few dominant directions.

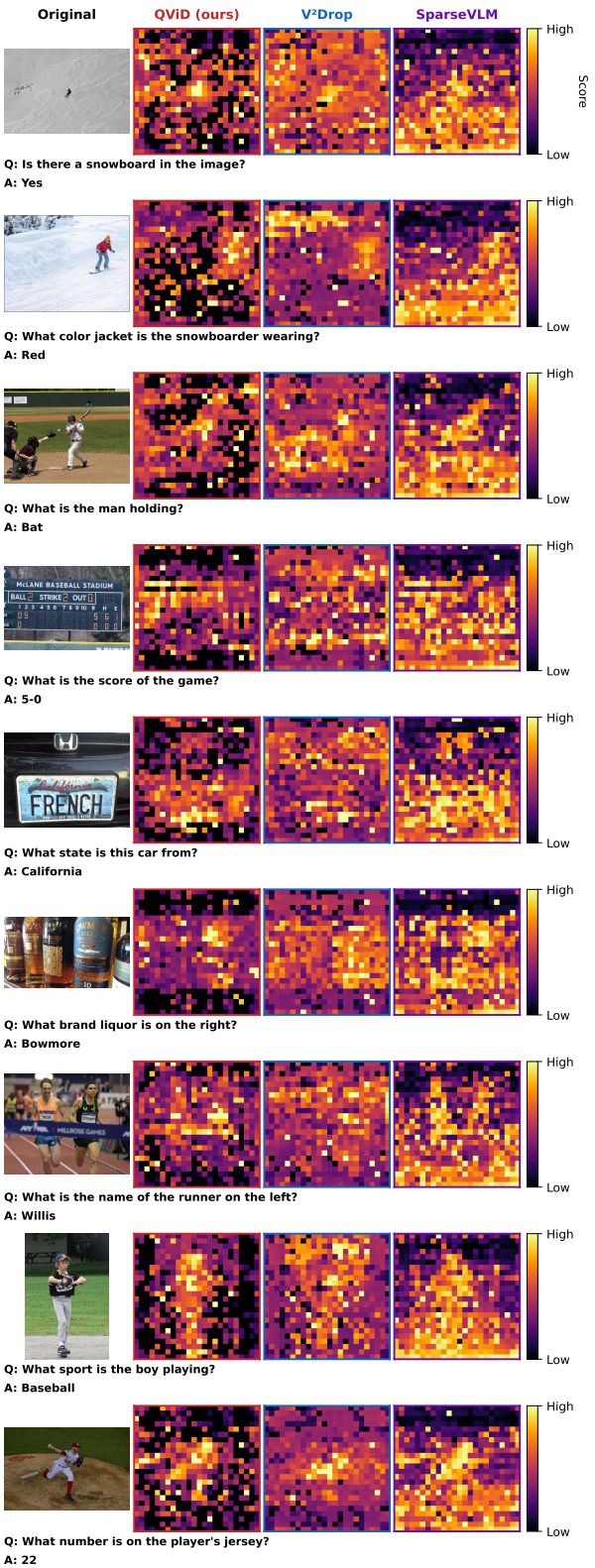

Figure 7. **Additional Qualitative Token-Score Visualizations.** We compare token-score maps from QViD, V^2Drop, and SparseVLM across multiple image-query pairs. Brighter regions indicate higher token scores. QViD tends to assign high scores to query-relevant visual regions, while the baseline methods often produce more diffuse score maps.

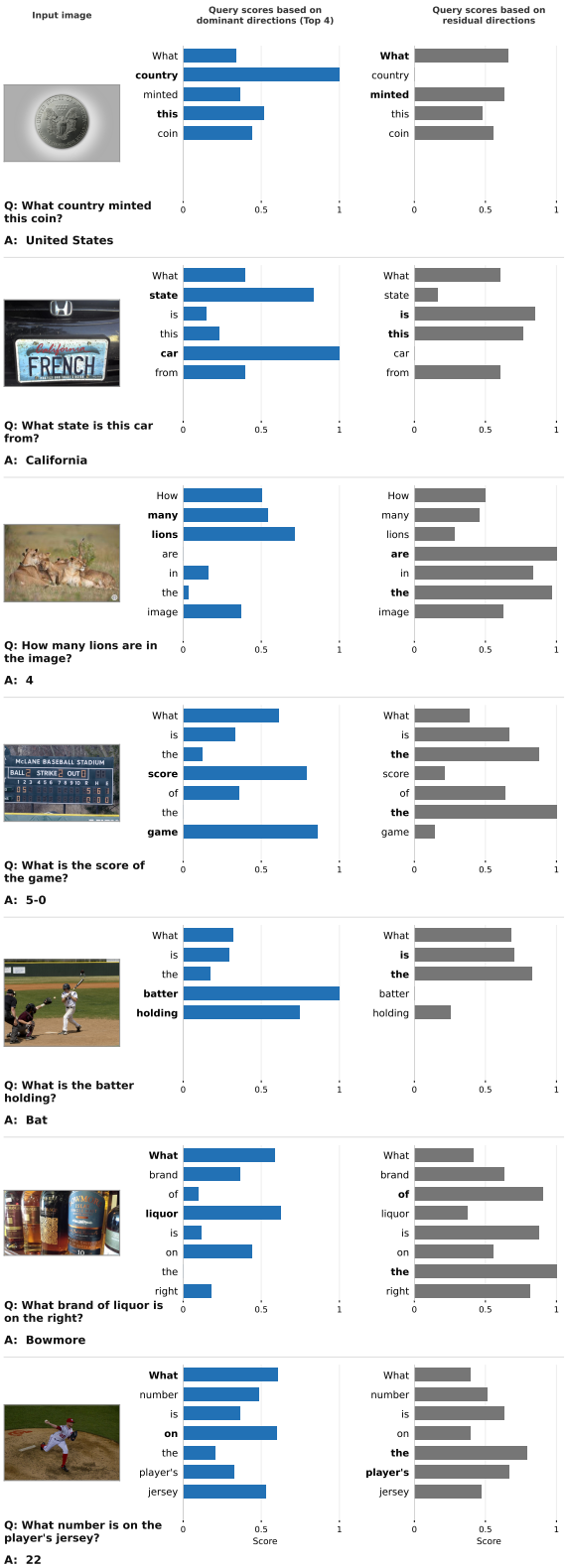

Figure 8. **Additional Query-Token Score Visualizations.** For each example, we compare query-token scores obtained from the dominant SVD components with scores from the residual components. The dominant components place higher weight on semantically informative query tokens, while residual components are less aligned with the task-relevant words.