TOWARD EMBEDDED VISION-LANGUAGE PERCEPTION FOR LONG-TERM AUTONOMOUS ROBOTS VIA TRAINING-FREE TOKEN PRUNING

Yvon Apedo[⋆] *Martyna Poreba*[†] *Michal Szczepanski*[†] *Samia Bouchafa*[⋆]

[⋆] IBISC, Université Paris-Saclay, Université d'Evry, F-91020, Evry-Courcouronnes, France
[†] Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

ABSTRACT

Vision-language models (VLMs) offer promising capabilities for semantic scene understanding in long-term autonomous robots. However, their deployment on embedded platforms remains challenging due to the high computational and memory cost induced by hundreds of visual tokens processed by large language models. In this work, we investigate the deployment potential of token pruning for resource-constrained robotic systems. Building upon SVD-Prune, a recently proposed token selection method, we analyze the impact of aggressive vision token reduction on computational cost, memory footprint, and inference latency. Experiments conducted with LLaVA-1.5-7B show that reducing the visual sequence from 576 to 16 tokens substantially lowers inference cost while preserving strong multimodal reasoning performance. We further discuss practical considerations for future deployment on NVIDIA Jetson platforms, including memory constraints, and TensorRT integration.

***Index Terms*—** Vision–Language Models, Token Pruning, Computational Efficiency, Computational complexity

1. INTRODUCTION

Autonomous robots operating for extended periods in complex outdoor environments require semantic understanding capabilities that go beyond object detection and scene recognition. Vision-language models (VLMs) have recently emerged as a promising solution by combining visual perception with language-based reasoning, enabling robots to answer questions about their surroundings, interpret complex scenes, and interact naturally with humans. Despite their capabilities, deploying VLMs on embedded robotic platforms remains challenging. Modern VLMs rely on large language models and process hundreds of visual tokens per image, resulting in substantial computational, memory, and energy requirements. These constraints are particularly critical on edge devices such as the NVIDIA Jetson family, where memory capacity, memory bandwidth, and power budgets are significantly more limited than on desktop GPUs. Among the various approaches proposed to improve VLM efficiency, vision token pruning has emerged as a particularly attractive direction. By reducing the number of visual tokens processed by the language model, token pruning can simultaneously decrease computational complexity, and inference latency without modifying the underlying VLM architecture.

In this paper, we investigate the deployment implications of aggressive vision token pruning for embedded vision-language perception. Using SVD-Prune [1] as a representative case study, we analyze the relationship between token reduction, computational cost, memory footprint, and inference latency. We further discuss practical challenges associated with deploying large VLMs on the NVIDIA Jetson Orin family of embedded platforms.

2. RELATED WORK

VLMs encode images into substantially more tokens than text, making visual representations the primary computational bottleneck. This imbalance stems from spatial redundancy and semantic sparsity in visual data, leading to increased memory usage, computational cost, and inference latency. To reduce the computational burden induced by visual tokens, numerous token pruning strategies have recently been proposed for VLMs. Existing approaches mainly rely on attention-based importance metrics, similarity measures, or adaptive pruning schedules to identify and remove redundant visual tokens. Representative examples include FastV [2], VisionZip [3], SparseVLM [4], PyramidDrop [5], HiRed [6] and TRIM [7]. In practice, these pruning methods are evaluated under moderate post-pruning token budgets, typically retaining between 64 and 128 visual tokens, where they achieve a favorable balance between efficiency and accuracy. More recently, low-rank representations have been explored as an alternative mechanism for identifying informative visual tokens. In particular, SVD-Prune [1] formulates token pruning as a low-rank approximation problem and uses SVD-based leverage scores to estimate token importance, achieving strong performance even under aggressive token reduction.

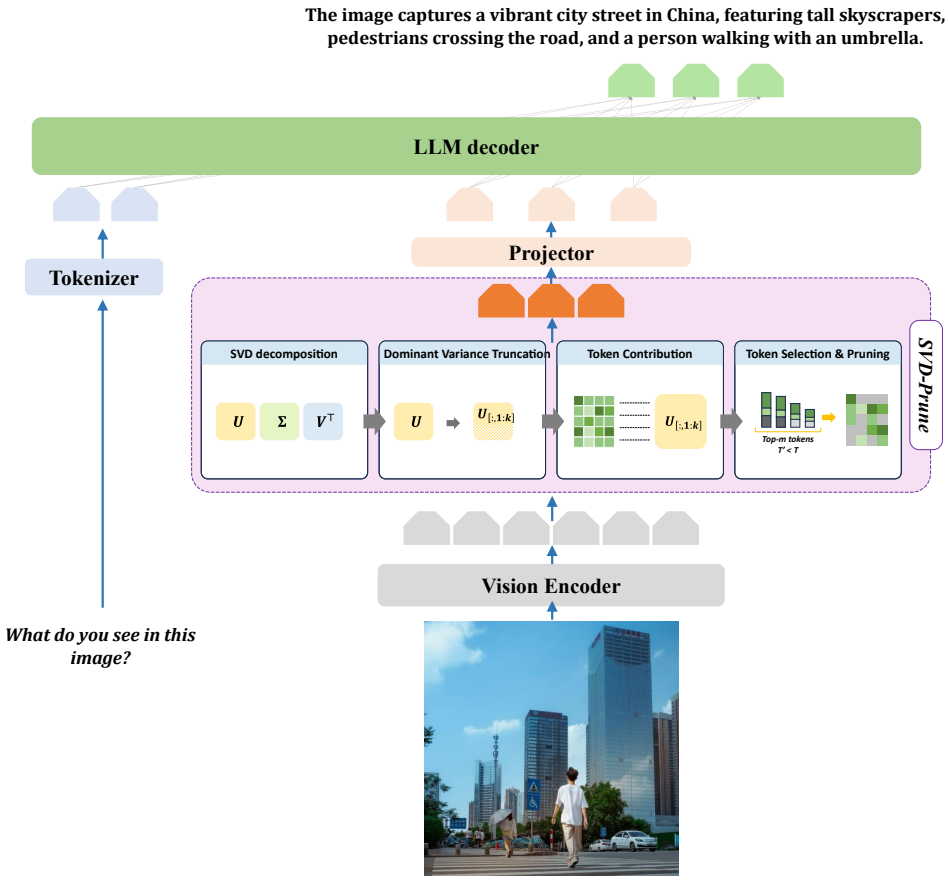

Fig. 1. Overview of SVD-Prune. The method performs outside-encoder vision token pruning by applying a global SVD decomposition to vision encoder outputs, estimating token importance via leverage scores, and selecting a compact subset of informative vision tokens before multimodal decoding.

3. SVD-PRUNE

SVD-Prune is a training-free vision token pruning framework that operates on the output representations of the vision encoder. As illustrated in Figure 1, the method identifies informative visual tokens through a global analysis of the visual feature space prior to multimodal decoding. Given the set of visual token embeddings produced by the vision encoder, SVD-Prune applies a Singular Value Decomposition (SVD) to uncover the dominant structures underlying the visual representation. These dominant components capture the principal variations present across the image and provide a compact summary of its most relevant visual content.

Unlike attention-based pruning approaches, which rely on local importance estimates and may be affected by positional biases, SVD-Prune considers the visual token set as a whole. This global perspective enables the identification of shared structures and redundancies across tokens, leading to a more holistic characterization of the image representation. Token importance is subsequently estimated according to each token's contribution to the dominant low-rank subspace extracted by the decomposition. Tokens that contribute most strongly to the preserved visual structure receive higher importance scores, whereas tokens associated with redundant or low-variance information receive lower scores. Based on these importance estimates, only a subset of highly informative tokens is retained and forwarded to the multimodal projector and language model. By preserving the dominant information content while substantially reducing sequence length, SVD-Prune enables more efficient vision-language inference, making it particularly attractive for deployment on resource-constrained robotic platforms.

4. EXPERIMENTS AND ANALYSIS

4.1. Experimental settings

In this study, we use LLaVA-1.5-7B [8] as the baseline vision-language model. All experiments are conducted using the native FP16 inference format. We evaluate the chosen token reduction strategy on GQA [9] and TextVQA [10]. While GQA

Table 1. Comparison on GQA and TextVQA under varying vision token budgets.

Methods	Venue	Location	GQA	TextVQA
Vanilla	NeurIPS'23	—	61.90	58.20
Retain 192 Tokens				
ToMe	ICLR'23	Encoder	54.30	52.10
FastV	ECCV'24	LLM	52.88	52.50
PyramidDrop	AAAI'25	LLM	57.30	<u>56.50</u>
Ours	—	Encoder	**59.88**	**57.24**
Retain 128 Tokens				
ToMe	ICLR'23	Encoder	52.40	49.10
FastV	ECCV'24	LLM	49.60	50.60
PyramidDrop	CVPR'25	LLM	57.10	**56.60**
VisionZip	CVPR'25	Encoder	<u>57.60</u>	55.80
Ours	—	Encoder	**58.70**	<u>56.14</u>
Retain 64 Tokens				
ToMe	ICLR'23	Encoder	48.60	45.30
FastV	ECCV'24	LLM	46.10	47.80
TRIM	arXiv'25	Encoder	50.90	50.00
PyramidDrop	CVPR'25	LLM	47.50	50.60
SparseVLM	ICLR'25	LLM	<u>53.70</u>	<u>53.40</u>
Ours	—	Encoder	**53.77**	**55.14**
Retain 32 Tokens				
ToMe	ICLR'23	Encoder	43.60	38.30
FastV	ECCV'24	LLM	41.50	42.50
SparseVLM	ICML'25	LLM	48.30	46.10
VisionZip	CVPR'25	Encoder	<u>51.80</u>	<u>53.10</u>
Ours	—	Encoder	**53.52**	**54.81**
Retain 16 Tokens				
Ours	—	Encoder	53.04	54.03

measures visual reasoning and scene-level understanding, TextVQA evaluates the ability to extract and reason over textual information embedded in images. Together, these tasks reflect key perception capabilities required by autonomous robots interacting with complex outdoor environments. Following standard LLaVA-1.5 settings, input images are resized to a resolution of 336×336, resulting in 576 vision tokens before pruning. All experiments are conducted using PyTorch on a single NVIDIA RTX 3080 16 GB GPU with an Intel Core i7-11800H CPU, which do not fully reflect the constraints of embedded platforms. In particular, the memory bandwidth, power envelope, and thermal limitations of edge devices such as the NVIDIA Jetson.

4.2. Comparison experiments

We compare SVD-Prune against representative encoder-side and decoder-side token pruning approaches under varying visual token budgets. As reported in Table 1, particular attention is given to aggressive compression regimes (32 and 16 retained tokens), which are especially relevant for deployment on resource-constrained platforms.

Across all evaluated token budgets, SVD-Prune consistently achieves the best or among the best performance. On GQA, the method maintains strong reasoning capabilities despite substantial sequence length reduction, achieving 59.88% accuracy with 192 retained tokens and remaining competitive even in the extreme low-token regime with 53.52% and 53.04% accuracy at 32 and 16 tokens, respectively. Similar trends are observed on TextVQA, where SVD-Prune preserves text-related visual information under aggressive pruning and reaches 54.81% and 54.03% accuracy when retaining only 32 and 16 visual tokens.

More importantly, the results indicate that reducing the visual sequence from 576 tokens to only a few dozen tokens leads to a relatively moderate accuracy degradation compared to the substantial reduction in computational complexity reported in the following section. This observation highlights the significant redundancy present in visual token representations and suggests that aggressive token pruning constitutes a promising strategy for efficient vision-language inference on embedded robotic platforms.

4.3. Computational Overhead Analysis

Table 2 analyzes the computational impact of vision token reduction. While the computational cost of the vision encoder remains constant, the costs of both the projector and the LLM scale linearly with the number of retained vision tokens. Consequently, aggressive vision token pruning yields substantial efficiency gains, reducing total FLOPs by 58.7%, 68.2%, 77.7%, 82.5%, and up to 84.8% when retaining 192, 128, 64, 32, and 16 tokens, respectively. In particular, reducing the token count from 576 to 16 lowers total computation from 3.45 T to 0.52 T FLOPs, highlighting that vision token count is the primary driver of inference cost.

4.4. Deployment-Oriented Efficiency Analysis on Commodity GPU Hardware

To assess the practical efficiency of SVD-Prune, we measure latency and memory usage by comparing the full model with its pruned version retaining only 16 tokens. As shown in Table 3, SVD-Prune reduces the number of vision tokens from 576 to 16 (a 36× reduction). This yields a substantial 4.18× reduction in per-image latency (from 930 ms down to 222.56 ms) and increases throughput from 1.06 FPS to 4.33 FPS (a 4.08× speedup). Peak VRAM usage also decreases

Table 2. FLOPs breakdown as a function of retained vision tokens.

Tokens	Vision [G]	Projector [G]	LLM [T]	Total [T]	Reduction [%]
576	190.600	12.080	3.250	3.450	0.00
192	190.600	4.030	1.230	1.430	58.7
128	190.600	2.680	0.903	1.100	68.2
64	190.600	1.340	0.576	0.770	77.7
32	190.600	0.671	0.413	0.604	82.5
16	190.600	0.336	0.332	0.523	84.8

Table 3. Efficiency analysis on RTX 3080 16G.

Method	Tokens	Latency [ms]	VRAM [GB]	FPS	Kernel time [ms]
Vanilla	576	930.00	14.11	1.06	670.03
SVD-Prune	16	222.56	13.82	4.33	338.34
Speedup	$\times$36.0	$\times$4.18$\downarrow$	$-$0.29	$\times$4.08$\uparrow$	$\times$1.98$\uparrow$

slightly by 0.29 GB, confirming the method's memory efficiency. These results demonstrate that SVD-based token pruning achieves significant acceleration with minimal overhead, making it highly suitable for deployment in compute- and memory-constrained scenarios.

5. DISCUSSION

5.1. Toward Deployment on NVIDIA Jetson Embedded Platforms

Even on recent edge hardware such as the NVIDIA Jetson Orin family, the deployment of a 7B-parameter VLM remains challenging due to tight memory and compute budgets. These Jetson platforms offer memory capacities ranging from 4 GB to 64 GB of unified LPDDR5. LLaVA-1.5-7B requires approximately 14 GB for the weights alone. This immediately rules out deployment in FP16 on all modules with 16 GB or less. With INT8 quantization the weight footprint drops to approximately 7 GB, making mid-range modules such as the Orin NX 16 GB viable, while more aggressive quantization up to INT4 ($\sim$3.5 GB) is necessary to target the most constrained devices. However, quantization alone is insufficient, as it reduces weights but does not address runtime memory, notably the KV cache, whose cost scales with sequence length and model depth. With SVD-Prune, reducing the number of vision tokens from 576 to 16 yields an approximately 36$\times$ reduction in the vision KV cache footprint, directly addressing this bottleneck and enabling more efficient deployment on memory-constrained devices. This highlights that efficient deployment of VLMs requires jointly addressing weight compression and sequence length reduction, motivating the use of complementary techniques such as quantization and token pruning.

5.2. ONNX export as an engineering challenge

Deploying LLaVA-1.5-7B on the Jetson platform requires building optimized TensorRT inference engines, which in turn depend on a valid ONNX representation of the model. Since the model already operates in FP16, the difficulty does not lie in a precision conversion but rather in the fundamental incompatibility between the ONNX format and the architectural characteristics of LLaVA. The first obstacle is a size limitation of the serialization format itself. ONNX encodes the entire computational graph, including all the weights, into a single protobuf message, which is subject to a hard limit of 2 GB. Direct ONNX export may appear to succeed but produces an invalid file that cannot be executed. While the ONNX specification does support an external data format that stores the weights in separate files, integrating this option reliably into a complex multi-component model is not straightforward. The second difficulty is specific to the LLaVA architecture and its dynamic control flow. The image-token splicing operation, which inserts projected vision embeddings into the text embedding sequence, involves conditional logic and variable-length tensor manipulations that cannot be traced cleanly by the ONNX exporter. Similarly, the KV-cache management differs between the prefill and the decode phases, introducing shape mismatches during tracing. These issues suggest that the model must be decomposed into separate subgraphs, each exported and optimized independently, and reassembled at runtime, which adds significant complexity to the deployment effort.

6. CONCLUSION

In this paper we investigated the potential of aggressive vision token reduction as an enabling technology for embedded vision-language perception. Using SVD-Prune as a representative training-free token pruning framework, we analyzed the impact of visual sequence length on multimodal reasoning performance, computational complexity, and runtime efficiency. Experimental results on GQA and TextVQA demonstrate that strong performance can be maintained even when the number of visual tokens is reduced from 576 to only 32 or 16 tokens.

Beyond accuracy, our analysis highlights the substantial efficiency gains enabled by token pruning. Reducing the visual sequence length significantly lowers the computational cost of the multimodal projector and language model, resulting in notable improvements in inference latency and throughput. These observations confirm that visual token redundancy constitutes an important opportunity for improving the deployability of large vision-language models.

At the same time, deploying VLMs on embedded platforms remains a broader systems challenge involving model compression, runtime optimization, and inference engine integration. While a full deployment on NVIDIA Jetson hardware is left for future work, our results suggest that combining aggressive token pruning with quantization techniques represents a promising direction toward real-time vision-language perception for long-term autonomous robotic systems.

Acknowledgments. This work was conducted within the NEUROKIT2E project, funded by the EU Horizon Europe programme (Grant Agreement No. 101112268).

7. REFERENCES

[1] Yvon Apedo, Martyna Poreba, Michal Szczepanski, and Samia Bouchafa, "Beyond attention scores: Svd-based vision token pruning for efficient vision-language models," 2026.

[2] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang, "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Cham, 2024, pp. 19–35, Springer.

[3] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia, "Visionzip: Longer is better but not necessary in vision language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19792–19802.

[4] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang, "Sparsevlm: Visual token sparsification for efficient vision-language model inference," in *International Conference on Machine Learning (ICML)*, 2025, Accepted to ICML 2025.

[5] Jiaqi Wang, Feng Wu, and Dahua Lin, "Conical visual concentration for efficient large vision-language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[6] Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji, "Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, vol. 39, pp. 1773–1781.

[7] Guan Song and Benyou Wang, "Less is more: A simple yet effective token reduction method for efficient multimodal llms," in *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, 2025, pp. 7614–7623.

[8] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds. 2023, vol. 36, pp. 34892–34916, Curran Associates, Inc.

[9] Drew A Hudson and Christopher D Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6700–6709.

[10] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach, "Towards vqa models that can read," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8317–8326.