

VPRUNE: EFFICIENT TRAINING-FREE PRE-LLM VISUAL TOKEN PRUNING

Guangchuan Lv¹, Dianxing Shi^{2*}, Dingjie Fu³

¹ Northeastern University, ² Beihang University, ³ Huazhong University of Science and Technology

ABSTRACT

Visual token pruning is a promising approach to reducing the inference cost of large vision-language models (LVLMs), yet aggressive token reduction often causes substantial performance degradation. We identify three key factors behind this degradation: text-guided selection bias, information loss from discarded tokens, and positional distortion caused by sequence compaction. Based on these observations, we propose **VPRune**, a training-free pre-LLM pruning framework consisting of visual-only diversity selection, similarity-guided token recycling, and position-preserving restoration. Experiments on FastVLM-1.5B across multiple vision-language benchmarks demonstrate that VPRune achieves a favorable accuracy–compression trade-off, with particularly pronounced advantages under aggressive compression. Furthermore, evaluations on edge-device show that VPRune effectively reduces end-to-end inference latency while maintaining superior task performance, demonstrating its practicality for resource-constrained LVLM deployment.

Index Terms— Vision-language models, visual token pruning, token reduction, efficient inference, edge AI

1. INTRODUCTION

Large vision-language models (LVLMs) achieve strong multimodal understanding capability by jointly processing visual and textual tokens [1, 2, 3]. However, visual representations often contain substantial redundancy, while long visual-token sequences increase inference latency and hinder deployment on resource-constrained edge devices. To this end, visual token pruning alleviates this cost by removing redundant visual tokens aggressively [4, 5], but such reduction often causes severe performance degradation, particularly when pruning is performed before the LLM.

Existing visual-token reduction methods generally improve efficiency by removing redundant tokens before the LLM or within early decoder layers [4, 6, 7, 8]. To identify informative tokens, they typically rely on visual redundancy, attention responses, or cross-modal relevance [5, 9]. However, many existing pruning methods are developed around token-rich LVLM settings, while increasingly efficient edge-oriented models are designed with compact multimodal repre-

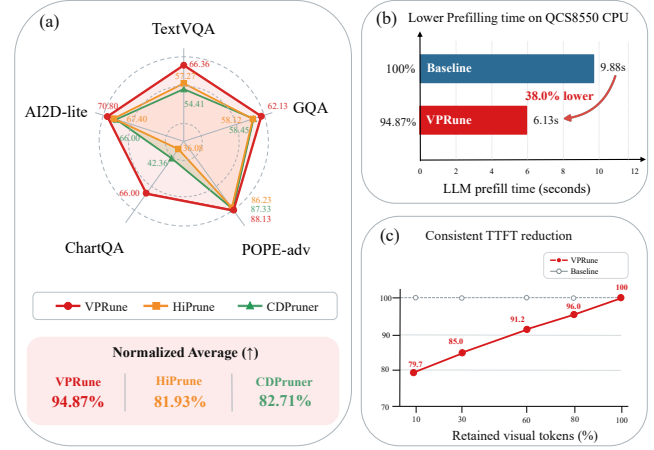


Fig. 1. Performance and edge efficiency of VPRune. At 30% token retention, VPRune achieves the highest normalized average performance of 94.87% in (a), while reducing LLM prefill latency by 38.0% on the QCS8550 CPU in (b); (c) further shows consistent end-to-end TTFT reductions across different visual-token budgets.

sentations for practical deployment [3, 10, 11]. In this regime, applying the same retention ratio yields a much tighter absolute token budget, making further pruning more sensitive to information loss and representation distortion.

To understand these degradations, we analyze pre-LLM pruning under tight token budgets and identify three empirical observations: **i)** text-conditioned relevance can over-concentrate token selection and reduce visual coverage, **ii)** discarded tokens are often redundant rather than uninformative and may still contain complementary visual evidence, and **iii)** sequence compaction reassigns retained tokens to consecutive position indices, distorting the spatial relationships seen by the pretrained LLM. Motivated by these observations, we propose **VPRune**, a training-free pre-LLM framework with three corresponding components. Specifically, we first introduce *Visual-only Diversity Selection* (VDS) to remove text conditioning and preserves visual diversity. We then employ *Similarity-guided Token Recycling* (STR) to transfer complementary information from discarded tokens to similar retained tokens. Moreover, we propose *Position-Preserving*

*Corresponding author: dianxingshi10@gmail.com

Restoration (PPR) to preserve their original raster order and position indices. Together, these components preserve visual evidence and positional structure under high compression without modifying the underlying LLM.

As highlighted in Fig. 1, VPRune provides a favorable accuracy–efficiency trade-off under aggressive compression. At 30% token retention, it achieves a normalized average performance of 94.87%, substantially outperforming representative methods. At the same retention ratio, VPRune reduces LLM prefill latency from 9.88 s to 6.13 s (38.0%) on a Qualcomm QCS8550 platform, while also providing consistent TTFT reductions across different token budgets.

Our contributions are: **i)** We analyze three key causes of performance degradation under aggressive visual-token pruning. **ii)** We propose VPRune, a training-free pre-LLM framework integrating VDS, STR, and PPR to advance low-budget token reduction performance. **iii)** Extensive experiments demonstrate strong accuracy under aggressive compression together with practical prefill and end-to-end latency reductions on a CPU-only edge platform.

2. RELATED WORK

Modern LVLMs typically employ a vision encoder, multimodal projector, and LLM to jointly process visual and textual representations [1, 2]. Efficient architectures such as FastVLM [3] reduce visual-processing overhead, while visual token reduction provides a complementary way to remove redundant representations. Existing methods prune tokens at different stages: HiPrune [6] and VisionZip [4] exploit visual structure or attention patterns, CDPruner [5] combines diversity-aware DPP selection with text-conditioned relevance, FlashVLM [9] performs efficient pre-LLM filtering, VScan [7] incorporates query-aware token reduction, and FastV [8] prunes tokens after early LLM processing. Despite their effectiveness, preserving visual content, positional consistency, and discarded information remains challenging under aggressive compression. VPRune addresses these issues entirely before LLM computation.

3. METHODOLOGY

3.1. Overview

As illustrated in Fig. 2, VPRune is a training-free pre-LLM visual token pruning framework designed for aggressive token budgets. Given the projected visual tokens $X = \{x_i\}_{i=1}^N$, where N denotes the number of visual tokens. VPRune consists of three stages. First, *Visual-only Diversity Selection* (VDS) removes the potentially misleading text-relevance bias and selects a compact yet diverse visual subset. Second, *Similarity-guided Token Recycling* (STR) recovers complementary information from discarded tokens by aggregating

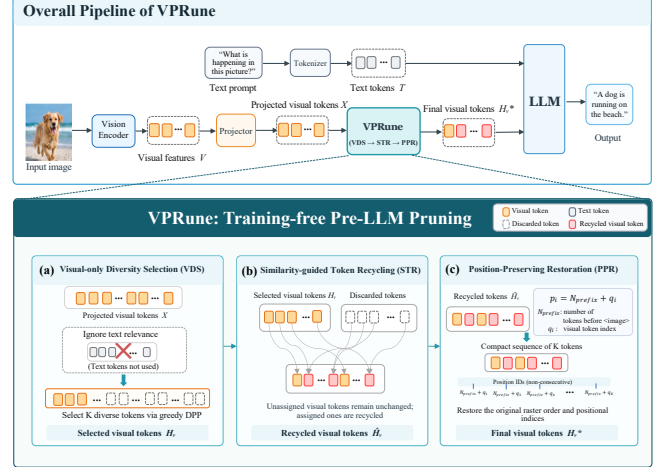


Fig. 2. Overview of VPRune. VDS selects diverse visual tokens, STR recycles complementary information from discarded tokens, and PPR restores their original positional structure before LLM inference.

them into their most similar retained tokens. Finally, *Position-Preserving Restoration* (PPR) restores the original raster order and positional indices of the retained tokens before they are fed into the LLM.

For a token retention ratio b , we retain $K = \text{round}(bN)$ visual tokens. In our setting, $N = 256$, and pruning is performed after the multimodal projector and before the LLM. Thus, the LLM operates on the reduced visual sequence.

3.2. Visual-only Diversity Selection

Let L denote the cosine-similarity-based visual diversity kernel and $D_r = \text{diag}(\hat{r}_1, \dots, \hat{r}_N)$ is the diagonal matrix of normalized text-relevance scores. The conditioned kernel is therefore $\tilde{L} = D_r L D_r$. For a retained subset S of size K , the corresponding DPP objective [12] can be decomposed as

$$\log \det(\tilde{L}_S) = \log \det(L_S) + 2 \sum_{i \in S} \log \hat{r}_i. \quad (1)$$

This decomposition reveals that text relevance contributes an explicit additive bias to the diversity objective. Under aggressive compression, visually complementary tokens may therefore be strongly suppressed because they receive low text-relevance scores. VDS removes this conditioning and performs selection solely according to visual diversity. The retained subset is obtained by

$$S^* = \arg \max_{|S|=K} \log \det(L_S). \quad (2)$$

We solve this objective with greedy MAP inference [13].

Table 1. Performance comparison on FastVLM-1.5B under different visual-token reduction ratios. Baseline denotes the full-token model; “Reduce (%)” is the percentage of visual tokens removed relative to it. “Avg.” is the mean of the five benchmark scores below normalized by their respective full-token baselines (Baseline = 100%). Best and second-best results among reduction methods in each column are highlighted in bold and underlined, respectively.

Method	TextVQA			GQA			POPE-adv			AI2D-lite			ChartQA			Avg.		
Reduce (%)	20	40	70	20	40	70	20	40	70	20	40	70	20	40	70	20	40	70
Baseline	70.58			63.53			87.97			73.00			77.32			100 (%)		
CDPruner	67.08	64.17	54.41	61.85	60.91	58.45	87.77	87.50	<u>87.33</u>	71.00	68.80	66.00	63.88	57.28	42.36	94.41	90.92	82.71
VScan	67.62	55.90	32.06	63.21	60.64	54.71	87.53	84.30	<u>77.57</u>	<u>72.20</u>	69.80	66.40	76.96	57.04	39.84	98.65	87.97	72.44
FastV	69.19	66.67	54.05	63.49	<u>62.97</u>	<u>58.69</u>	<u>88.10</u>	<u>87.93</u>	83.23	71.20	70.00	66.40	75.64	<u>73.64</u>	<u>54.36</u>	<u>98.70</u>	<u>96.93</u>	<u>84.97</u>
HiPrune	69.17	<u>67.73</u>	57.27	62.33	61.15	58.12	88.23	88.13	86.23	71.40	<u>70.40</u>	<u>67.40</u>	67.40	61.08	36.08	96.28	93.57	81.93
VisionZip	56.82	55.04	39.78	59.29	59.17	57.64	87.77	87.63	86.43	65.60	66.40	65.20	35.12	35.28	29.12	81.78	81.46	74.46
FlashVLM	<u>69.58</u>	67.54	<u>57.87</u>	62.39	60.85	57.17	88.07	87.77	85.93	71.20	70.00	66.40	68.12	62.08	35.20	96.51	93.49	81.23
VPRune (Ours)	70.57	69.64	66.36	<u>63.29</u>	63.07	62.13	88.00	<u>87.93</u>	88.13	72.40	70.80	70.80	<u>76.20</u>	74.08	66.00	99.47	98.14	94.87

3.3. Similarity-guided Token Recycling

Directly removing unselected tokens may discard complementary visual information, especially under tight token budgets. STR therefore recycles discarded features into their most similar retained tokens. For each discarded token $i \in D$, we assign it to $j^*(i) = \arg \max_{j \in S} \cos(x_i, x_j)$, reusing the similarities already computed in the DPP kernel. Let $G_j = \{i \in D \mid j^*(i) = j\}$ denote the discarded tokens assigned to retained token j . Their contribution is weighted according to

$$w_{ij} = \frac{\exp(L_{ij}/\tau)}{\sum_{i' \in G_j} \exp(L_{i'j}/\tau)}. \quad (3)$$

Rather than introducing additional tokens, STR directly updates the retained representation as

$$\tilde{x}_j = (1 - \gamma)x_j + \gamma \sum_{i \in G_j} w_{ij}x_i, \quad (4)$$

where γ controls the amount of recycled information. If no discarded token is assigned to j , i.e., $G_j = \emptyset$, we simply keep $\tilde{x}_j = x_j$. Thus, STR recovers complementary information without increasing the number of tokens passed to the LLM.

3.4. Position-Preserving Restoration

Conventional token pruning typically compacts the retained visual sequence and reassigns consecutive position indices, altering the positional relationships encoded by RoPE-based LLMs [14]. Let the retained tokens correspond to original raster indices $S = \{q_i\}_{i=1}^K$ with $q_1 < \dots < q_K$. PPR first restores their original sequence order and assigns each retained token the position

$$p_i^{\text{vis}} = N_{\text{prefix}} + q_i, \quad (5)$$

instead of the compact position $N_{\text{prefix}} + i$. Following GAP [15], text tokens after the image start from $N_{\text{prefix}} + N$, preserving the original visual-token span. Importantly, the retained embeddings remain a compact K -token sequence; gaps exist only in the assigned position indices rather than as empty token slots. This preserves the original raster order and relative positional distances without reconstructing a length- N visual sequence.

4. EXPERIMENTS

4.1. Experimental Setup

Model and Baselines. We evaluate VPRune on the frozen FastVLM-1.5B model [3] and compare it with six representative visual-token reduction methods: CDPruner [5], VScan [7], FastV [8], HiPrune [6], VisionZip [4], and FlashVLM [9].

Benchmarks. We evaluate five complementary benchmarks: GQA [16] for compositional reasoning, TextVQA [17] for text-rich understanding, POPE-adv [18] for object hallucination, a 500-sample subset of AI2D [19] (AI2D-lite) for diagram understanding, and ChartQA [20] for chart reasoning.

Implementation Details. FastVLM-1.5B uses FastViT-HD [21] and a Qwen2 LLM. Inputs are padded to 1024×1024 , yielding 256 visual tokens; VPRune prunes after the projector and before the LLM. We report accuracy at $b \in \{0.3, 0.6, 0.8\}$ and additionally use $b = 0.1$ for efficiency. Baselines are matched to the target ratios. All runs use greedy decoding (batch size 1) without fine-tuning; VPRune sets $\gamma = 0.3$ and $\tau = 0.1$.

Edge Platform. Latency is measured on a Qualcomm QCS8550-based development platform using CPU-only inference, with GPU and NPU acceleration disabled. The QCS8550 integrates a 64-bit Kryo CPU with a maximum clock frequency of 3.2GHz, providing a representative

resource-constrained platform for evaluating practical edge inference.

4.2. Main Results

As shown in Table 1, VPRune achieves the highest normalized average performance at all reported reduction ratios, retaining 99.47%, 98.14%, and 94.87% of full-token performance at 20%, 40%, and 70% reduction, respectively. More importantly, its margin over the second-best method grows from 0.77 and 1.21 percentage points at 20% and 40% reduction to 9.90 points over FastV at 70% reduction. At this setting, VPRune ranks first on all five benchmarks. The largest gains occur on TextVQA and ChartQA, where it exceeds the corresponding second-best results by 8.49 and 11.64 raw points. These tasks rely heavily on fine-grained textual or structured visual evidence and are therefore more sensitive to information removed by aggressive pruning. This pattern is consistent with VPRune’s design: VDS and STR preserve distributed and complementary visual evidence, while PPR maintains spatial relationships that are particularly important for structured tasks such as ChartQA. The smaller gap on POPE-adv, whose scores remain near the full-token baseline for several methods, further suggests that the main benefit appears on content-dense reasoning tasks rather than from a uniform score shift.

4.3. Ablation Study

We conduct ablations at a token retention ratio of $b = 0.3$. As shown in Fig. 3, removing text-guided relevance yields a substantial gain on TextVQA and a modest improvement on GQA. Position-preserving restoration further improves both benchmarks, while similarity-guided token recycling provides consistent additional gains by recovering complementary information from discarded tokens.

We further study the effect of the recycling strength γ on TextVQA and GQA at $b = 0.3$. Performance remains relatively stable over moderate values of γ but degrades when excessive discarded information is introduced. We therefore set $\gamma = 0.3$, which provides the best overall trade-off across the two benchmarks.

4.4. Efficiency Analysis

We evaluate practical inference efficiency on the QCS8550 CPU-only edge platform. As shown in Fig. 4, VPRune performs pruning after the multimodal projector and therefore maintains near-baseline vision-encoder latency, while VScan, VisionZip, and HiPrune incur higher vision-side overhead.

More importantly, reducing the visual sequence before LLM inference provides consistent end-to-end benefits. VPRune reduces TTFT from 25.551 s for the full-token baseline to 20.365 s at $b = 0.1$, corresponding to a 20.30% reduction. The reductions remain 14.97%, 8.82%, and 4.02%

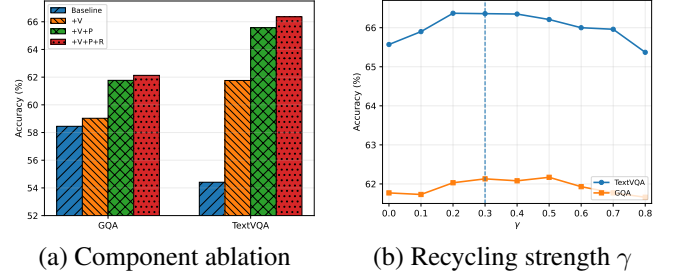


Fig. 3. Ablation studies under 30% token retention. (a) Cumulative ablation of V, P, and R improves performance, where V, P, and R denote visual-only diversity selection, position-preserving restoration, and similarity-guided token recycling, respectively. (b) $\gamma = 0.3$ gives the best overall trade-off on TextVQA and GQA.

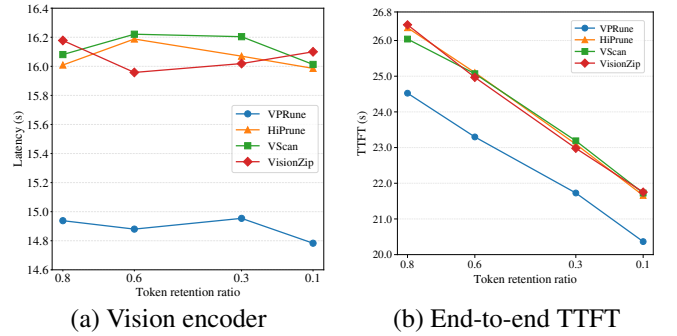


Fig. 4. Efficiency on the QCS8550 CPU-only edge platform. VPRune maintains near-baseline vision-encoder latency in (a) and consistently reduces end-to-end TTFT across token-retention ratios in (b).

at $b = 0.3, 0.6$, and 0.8 , respectively. Its TTFT remains comparable to the fastest competing methods while maintaining substantially higher accuracy under aggressive compression, demonstrating a favorable accuracy–latency trade-off for edge deployment.

5. CONCLUSION

In this work, we investigate the performance degradation of visual token pruning under aggressive compression and propose **VPRune**, a training-free pre-LLM framework for efficient LVLM inference. VPRune addresses three key limitations of low-budget pruning through visual-only diversity selection, similarity-guided token recycling, and position-preserving restoration. Experiments on FastVLM-1.5B across five vision-language benchmarks demonstrate that VPRune achieves a favorable accuracy–compression trade-off, particularly under tight token budgets. CPU-only edge evaluations further show reduced end-to-end latency while maintaining strong task performance, supporting efficient resource-constrained LVLM deployment.

6. REFERENCES

- [1] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi, “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in *ICML*, 23–29 Jul 2023, vol. 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742.
- [2] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee, “Visual instruction tuning,” in *NeurIPS*, 2023, vol. 36, pp. 34892–34916.
- [3] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari, “Fastvlm: Efficient vision encoding for vision language models,” in *CVPR*, June 2025, pp. 19769–19780.
- [4] 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 *CVPR*, June 2025, pp. 19792–19802.
- [5] Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang, “Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms,” in *NeurIPS*, 2025, vol. 38, Main Conference, pp. 25438–25468.
- [6] Jizhihui Liu, Guangdao Zhu, Feiyi Du, Niu Lian, Jun Li, Bin Chen, Weili Guan, and Yaowei Wang, “HiPrune: Hierarchical attention for efficient token pruning in vision-language models,” in *ACL Findings*, July 2026, pp. 3274–3291.
- [7] Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu, “VScan: Rethinking visual token reduction for efficient large vision-language models,” *TMLR*, 2026.
- [8] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang, “An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models,” in *ECCV*, 2024, pp. 19–35.
- [9] Kaitong Cai, Jusheng Zhang, Jing Yang, Yijia Fan, Pengtao Xie, Jian Wang, and Keze Wang, “Flashvlm: Text-guided visual token selection for large multimodal models,” 2025.
- [10] Xudong Lu, Yinghao Chen, Cheng Chen, Hui Tan, Boheng Chen, Yina Xie, Rui Hu, Guanxin Tan, Renshou Wu, Yan Hu, Yi Zeng, Lei Wu, Liuyang Bian, Zhaoxiong Wang, Long Liu, Yanzhou Yang, Han Xiao, Aojun Zhou, Yafei Wen, Xiaoxin Chen, Shuai Ren, and Hongsheng Li, “Bluelm-v-3b: Algorithm and system co-design for multimodal large language models on mobile devices,” in *CVPR*, June 2025, pp. 4145–4155.
- [11] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang, “Token-packer: Efficient visual projector for multimodal LLM,” *Int. J. Comput. Vis.*, vol. 133, no. 10, pp. 6794–6812, 2025.
- [12] Alex Kulesza and Ben Taskar, “Determinantal Point Processes for Machine Learning,” *Foundations and Trends in Machine Learning*, vol. 5, pp. 123–286, 2012.
- [13] Laming Chen, Guoxin Zhang, and Eric Zhou, “Fast greedy map inference for determinantal point process to improve recommendation diversity,” in *NeurIPS*, 2018, vol. 31.
- [14] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu, “Roformer: Enhanced transformer with rotary position embedding,” *Neuro-computing*, vol. 568, pp. 127063, 2024.
- [15] Tzu-Chun Chien, Chieh-Kai Lin, Shiang-Feng Tsai, Ruei-Chi Lai, Hung-Jen Chen, and Min Sun, “Grounding-aware token pruning: Recovering from drastic performance drops in visual grounding caused by pruning,” 2025.
- [16] Drew A. Hudson and Christopher D. Manning, “Gqa: A new dataset for real-world visual reasoning and compositional question answering,” in *CVPR*, June 2019.
- [17] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach, “Towards vqa models that can read,” in *CVPR*, June 2019.
- [18] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen, “Evaluating object hallucination in large vision-language models,” in *EMNLP*, Dec. 2023, pp. 292–305.
- [19] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi, “A diagram is worth a dozen images,” in *ECCV*, 2016, pp. 235–251.
- [20] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque, “ChartQA: A benchmark for question answering about charts with visual and logical reasoning,” in *ACL*, May 2022, pp. 2263–2279.
- [21] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan, “Fastvit: A fast hybrid vision transformer using structural reparameterization,” in *ICCV*, October 2023, pp. 5785–5795.