Are We Using the Right Benchmark: An Evaluation Framework for Visual Token Compression Methods

Anonymous ACL submission

Abstract

Recent endeavors to accelerate inference in Multimodal Large Language Models (MLLMs) have primarily focused on visual token compression. The effectiveness of these methods is typically assessed by measuring the accuracy drop on established benchmarks, comparing model performance before and after compression. However, these benchmarks are originally designed to assess the perception and reasoning capabilities of MLLMs, rather than to evaluate compression techniques. As a result, directly applying them to visual token compression introduces a task mismatch. Strikingly, our investigation reveals that ***simple image downsampling consistently outperforms many advanced compression methods across multiple widely used benchmarks***. Through extensive experiments, we make the following observations: (i) Current benchmarks are noisy for the visual token compression task. (ii) Downsampling is able to serve as a data filter to evaluate the difficulty of samples in the visual token compression task. Motivated by these findings, we introduce VTC-Bench, an evaluation framework that incorporates a data filtering mechanism to denoise existing benchmarks, thereby enabling fairer and more accurate assessment of visual token compression methods. The code will be released to facilitate future research.

1 Introduction

Multimodal Large Language Models (MLLMs) have shown impressive abilities in understanding, reasoning, and generating content across vision and language, enabling applications such as embodied AI (Yin et al., 2024; Fu et al., 2025; Cheng et al., 2025; Yang et al., 2025c). However, their efficiency is often constrained by the high computational cost of processing images, particularly at high resolutions (Liu et al., 2025). This bottleneck arises because visual tokens, derived from image patches, typically far outnumber textual tokens, leading to substantial memory consumption

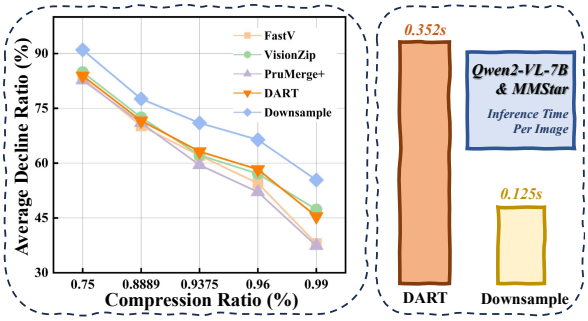

Figure 1: (a) Average Decline Ratio (ADR) of five visual token compression methods on eight benchmarks (Model: Qwen2-VL-7B; Benchmark: as shown in Table 1; Device: 1 A800)). (b) Inference time per image comparison of DART and Downsample (Model: Qwen2-VL-7B; Benchmark: MMstar; Compression Ratio: 0.75; Device: 1 A800).

and inference latency (Wang et al., 2025; Chen et al., 2025; Wen et al., 2025c). To mitigate this issue, numerous visual token compression methods have been proposed to reduce redundancy while retaining essential visual information (Yang et al., 2025a; Xing et al., 2024; Wen et al., 2025a; Xiong et al., 2025).

Yet, these methods are typically evaluated on general MLLM benchmarks (Li et al., 2024c), which are not designed for compression, therefore failing to provide appropriate evaluation criteria. Thus, in this paper, we uncover a surprising finding: as in Figure 1, ***simple image downsampling consistently outperforms many advanced compression methods across multiple widely used benchmarks.*** This suggests that current evaluation frameworks don't adequately capture the challenges inherent in visual token compression.

To investigate this, we conduct a comprehensive empirical study comparing multiple state-of-the-art visual token compression methods against a simple downsampling baseline across eight widely used benchmarks. Based on the study results in Table 1 and 2, two crucial findings are concluded: ①

The counterintuitive phenomenon mentioned above generally exists in popular benchmarks, proving that *current benchmarks are noisy for the visual token compression task*. ② The correct sample group under downsampling methods has achieved significantly better accuracy than the incorrect sample group under downsampling methods across various compression methods and benchmarks, proving that *downsampling can serve as a data filter to evaluate the difficulty of samples upon the visual token compression task.*

Based on these findings, we propose VTC-Bench, a new evaluation framework specifically designed to optimize and denoise current existing benchmarks, aiming to evaluate visual token compression methods fairly. By explicitly distinguishing between "simple" and "difficult" samples through downsampling, VTC-Bench adaptively and fairly selects "difficult" samples that satisfy the requirements of evaluating visual token compression methods.

Overall, our contributions are threefold: ① We identify and validate the data noise in existing MLLM benchmarks on the visual token compression task. ② We introduce a data filtering mechanism using downsampling as a discriminator to categorize benchmark samples by difficulty. ③ We propose VTC-Bench, the first evaluation framework tailored for fairly evaluating visual token compression methods, aiming to foster more meaningful progress in this emerging field.

2 Related Work

2.1 Visual Token Compression for MLLMs

Since visual tokens typically outnumber text tokens in MLLMs, compressing visual tokens has emerged as a promising strategy to accelerate inference (Liu et al., 2025). Leveraging the inherent redundancy in visual tokens, a variety of *training-free* methods have been proposed. FastV (Chen et al., 2024a), the first to explore visual token compression in MLLMs, prunes redundant tokens based on their average attention scores. Building on this idea, SparseVLM (Zhang et al., 2025) introduces a recycling strategy to achieve more compact and flexible compression. Other methods, such as PyramidDrop (Xing et al., 2024), FiCoCo-V (Han et al., 2024), and MustDrop (Liu et al., 2024a), divide the compression process into multiple stages, enabling more precise identification of redundant tokens. In contrast, DART (Wen et al., 2025b) departs from importance-based selection entirely and instead prioritizes token duplication as a key criterion, achieving surprisingly strong compression performance. Similarly, G-Prune (Jiang et al., 2025) identifies critical tokens through a graph-based perspective. Beyond these, GreedyPrune (Pei et al., 2025) and ToDRE (Li et al., 2025) cast token compression as an optimization problem and employ greedy algorithms to search for efficient pruning strategies. However, as in Sec. 3, we have a surprising observation: across most MLLM benchmarks, these sophisticated visual token compression methods under-perform compared to simply reducing the original image resolution, which motivates a deeper investigation into the underlying causes.

2.2 MLLM Benchmarks

Existing MLLM benchmarks primarily focus on areas such as perception and reasoning (Li et al., 2024c). For example, MME (Yin et al., 2024), MMBench (Liu et al., 2024b), SEED-Bench (Li et al., 2024b), and MM-Vet (Yu et al., 2023, 2024) provide broad perception-focused evaluations of MLLMs' visual understanding. In parallel, domain benchmarks target specific applications such as autonomous driving (Sima et al., 2024; Qian et al., 2024) and remote sensing (Muhtar et al., 2024). For visual token compression in MLLMs, only one benchmark currently exists: EffiVLM (Wang et al., 2025). It offers a unified framework for benchmarking training-free acceleration methods but relies on existing datasets (e.g., DocVQA (Mathew et al., 2021), ChartQA (Masry et al., 2022)) rather than data tailored to token compression. Building on data-driven insights into compression behavior, we introduce **VTC-Bench**, the first dedicated, challenging evaluation framework for visual token compression in MLLMs. We aim for VTC-Bench to catalyze new research and insights, enabling fair comparisons and sharper evaluations of token-compression methods.

3 Experiments & Findings

3.1 Motivation

Some of the recent MLLMs, such as Qwen2-VL (Wang et al., 2024) and Qwen2.5-VL (Bai et al., 2025), natively support inputs of varying resolutions. A trivial yet efficient method to handle high-resolution images is to simply downsample them to a lower resolution, effectively using naive pixel sampling as a form of compression. However, as

Table 1: Comparison of Advanced Token Compression Methods and Downsampling on Qwen2-VL-7B. ADR refers to the average decline ratio, which is the average value of the decline ratio of each benchmark.

Method	GQA	MMB	MMB^{CN}	MME	POPE	MMStar	OCRBench	ChartQA	ADR
Qwen2-VL-7B	*Upper Bound. All Tokens (100%)*								
Vanilla	62.3	78.9	78.0	2306	88.4	57.1	80.7	81.6	100.0
Qwen2-VL-7B	*Token Reduction (↓ 75.00%)*								
+ FastV	57.0	73.7	73.1	2083	84.5	44.6	42.0	58.1	83.2
+ VisionZip	58.6	71.1	70.5	2062	87.1	47.2	42.1	**66.9**	84.9
+ PruMerge+	**59.4**	72.1	72.0	2044	**87.2**	48.0	33.9	56.2	82.7
+ DART	56.9	72.5	70.2	2066	84.7	47.2	52.5	52.7	83.9
+ Downsample	59.2	**75.0**	**73.8**	**2259**	86.2	**50.1**	**64.9**	65.0	**91.0**
Qwen2-VL-7B	*Token Reduction (↓ 88.89%)*								
+ FastV	52.3	65.0	65.5	1854	77.4	40.3	25.9	32.9	70.2
+ VisionZip	53.3	62.9	63.0	1820	83.6	40.2	25.1	**48.4**	72.5
+ PruMerge+	54.8	62.2	61.3	1806	**84.3**	38.4	22.2	44.2	71.0
+ DART	51.9	61.3	61.8	1915	80.5	39.8	41.0	30.8	71.6
+ Downsample	**55.5**	**69.0**	**70.2**	**2127**	82.9	**44.0**	**48.8**	24.8	**77.6**
Qwen2-VL-7B	*Token Reduction (↓ 93.75%)*								
+ FastV	49.0	57.1	57.9	1684	74.9	37.5	18.7	20.6	62.1
+ VisionZip	49.0	54.8	54.0	1704	**80.2**	35.2	15.9	28.0	62.2
+ PruMerge+	48.7	48.4	48.1	1679	79.2	33.2	14.4	**30.0**	59.5
+ DART	49.2	53.4	54.0	1786	78.1	33.6	33.7	19.2	63.2
+ Downsample	**52.6**	**66.4**	**66.8**	**1994**	79.5	**40.9**	**40.3**	12.7	**71.0**
Qwen2-VL-7B	*Token Reduction (↓ 96.00%)*								
+ FastV	46.1	43.9	46.6	1589	72.4	33.6	14.4	15.8	54.5
+ VisionZip	46.4	49.5	50.0	1628	77.8	33.4	12.0	19.4	57.1
+ PruMerge+	45.0	39.1	40.9	1544	74.0	30.5	10.5	**20.9**	52.1
+ DART	45.6	47.9	48.2	1701	74.7	31.7	29.3	16.6	58.3
+ Downsample	**50.1**	**62.0**	**61.4**	**1938**	**78.8**	**37.5**	**32.3**	11.7	**66.4**
Qwen2-VL-7B	*Token Reduction (↓ 99.00%)*								
+ FastV	38.2	23.9	24.5	1189	55.0	26.1	5.8	11.9	38.0
+ VisionZip	41.9	40.5	40.5	1335	65.5	30.8	4.9	12.8	47.3
+ PruMerge+	39.0	23.7	24.4	1165	51.6	25.7	3.5	**13.9**	37.4
+ DART	40.5	30.8	30.7	1346	60.0	28.8	**23.2**	11.8	45.4
+ Downsample	**43.5**	**51.6**	**51.9**	**1589**	**72.8**	**33.8**	13.2	12.1	**55.4**

shown in Sec. 2.1, most token compression methods for MLLMs choose to adaptively drop useless tokens or merge similar tokens instead of directly downsampling the original image, which should be more intelligent and reasonable methods. While in recent works (Yang et al., 2025b), it is surprising that *image downsampling exceeds other sophisticated methods under some settings.* In order to further investigate the causes of this anomalous phenomenon, we decide to comprehensively compare the results of the downsampling methods with other methods under various settings.

3.2 Experiments Setup

Before conducting experiments, it is crucial to choose a suitable MLLM for achieving the downsampling method. Most MLLMs only support fixed-resolution inputs, which makes it impossible to achieve the downsampling method. In other words, for such MLLMs, no matter which resolution the original image is downsampled to, the image will finally be resized to a fixed resolution, making the downsampling meaningless. Considering that Qwen2-VL (Wang et al., 2024) and Qwen2.5-VL (Bai et al., 2025), based on the naive dynamic resolution mechanism and M-RoPE techniques, are the open-source MLLMs closest to realizing the concept of allowing arbitrary resolution inputs, we choose Qwen2-VL in our comparison experiments, which supports the downsampling method the best. In order to ensure that downsampling occurs at the original resolution as much as possible without adding extra resizing operations, we set Qwen2-VL's max pixels and min pixels to 2408448 and 3136. In this case, only a few extremely high-resolution images will be resized before downsampling to ensure sufficient GPU memory.

To guarantee comprehensive experiments, we select four typical token compression methods(FastV (Chen et al., 2024a), VisionZip (Yang et al., 2025a), PruMerge+ (Shang et al., 2024), and DART (Wen et al., 2025b)) with the token compression ratio set to 75.00%, 88.89%, 93.75%, 96.00%, and 99.00%. For the token compression ra-

tio C, the downsampling method applies an equivalent downsampling ratio D for fairness. The rule is shown in Eq. 1. Moreover, we choose eight popular benchmarks, including six general benchmarks (GQA (Hudson and Manning, 2019), MMBench_EN (Liu et al., 2024b), MMBench_CN (Liu et al., 2024b), MME (Yin et al., 2024), POPE (Li et al., 2023), and MMStar (Chen et al., 2024b)) and two resolution-sensitive OCR benchmarks (OCRBench (Liu et al., 2024c), and ChartQA (Masry et al., 2022)).

$$\frac{1}{D^2} \times 100\% = 1 - C \tag{1}$$

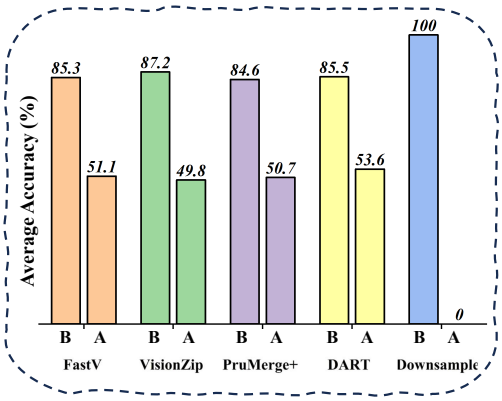

Figure 2: Comparison of advanced token compression methods and downsampling on Qwen2-VL-7B by groups at 75% compression.

3.3 Results Analysis

Comparison between Different Methods: Across a wide range of compression ratios and general-purpose benchmarks, naive image downsampling achieves superior performance compared to sophisticated token compression methods in most conditions. For instance, at 93.75% compression, downsampling achieves 66.4% on MMBench, outperforming the best advanced method, DART, by a 24.3% relative improvement. Similarly, on GQA, downsampling maintains a consistent lead across all compression ratios. The results verify a basic phenomenon in the field of visual token compression: *a substantial portion of samples in general-purpose benchmarks can be correctly answered using only low-resolution global information, without requiring the fine-grained visual details that advanced methods strive to preserve.*

Comparison between different compression ratios: As compression becomes increasingly aggressive (96.00% and 99.00%), all sophisticated token compression methods experience performance degradation, while image downsampling demonstrates remarkably graceful degradation. At 99.00% compression, downsampling maintains a score of 51.6% on MMBench, while FastV and PruMerge+ decrease to approximately 24%. The results further verify the phenomenon above: *in the existing general-purpose benchmarks, image downsampling can fully meet the acceleration requirements for most samples.*

Comparison between Different Tasks: On tasks requiring fine-grained visual understanding—particularly chart comprehension—we observe a reversal of the phenomenon mentioned above. At moderate compression ratios (93.75% and 88.89%), VisionZip and FastV outperform image downsampling on ChartQA by significant margins. This divergence is highly informative: while image downsampling uniformly preserves global information at the expense of local details, the sophisticated compression methods can selectively retain text regions and numeric values that are critical for chart understanding, which can be considered difficult to compress. Thus, a deeper observation of the above phenomenon can be concluded:*the sophisticated token compression methods demonstrate the expected effectiveness in tasks that require fine-grained visual understanding*.

The comparisons across methods, compression ratios, and tasks provide compelling evidence that current benchmarks contain a substantial simplicity bias. The performance advantage of image downsampling emerges not from its sophistication but from its ability to adequately address samples that don't require fine-grained visual understanding—precisely the samples that dominate current benchmarks. Thus, based on the experimental results and the comparisons, we propose a well-founded hypothesis in Section 3.4.

3.4 Hypothesis

In real life, if the difficulty of an exam is much lower than that of students, then students' grades will be chaotic, mainly manifested in the confusion of good students' and bad students' grades. In the field of visual token compression, there is a general reliance on existing benchmarks, without ever considering whether these data are suitable for the visual token compression task. Thus, we propose a bold hypothesis: ***Some data in the existing benchmarks is overly simplistic, leading to the unreasonable phenomenon that even the simplest***

Table 2: Comparison of advanced token compression methods and downsampling on Qwen2-VL-7B by groups.

Method	GQA	MMB	MMB^{CN}	MME	POPE	MMStar	OCRBench	ChartQA	Average
Group B	*Token Reduction (↓ 75.00%)*								
+ FastV	87.6	95.9	95.8	96.7	94.8	76.0	57.2	78.1	85.3
+ VisionZip	91.2	93.8	93.6	95.3	96.8	81.4	58.1	87.3	87.2
+ PruMerge+	91.9	95.1	94.6	95.9	97.5	82.3	46.2	73.6	84.6
+ DART	88.1	94.9	94.6	94.9	94.5	77.7	70.2	69.0	85.5
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 75.00%)*								
+ FastV	57.8	45.2	56.5	78.9	65.4	41.0	29.1	35.0	51.1
+ VisionZip	59.3	42.4	42.2	54.9	72.5	45.9	29.6	51.2	49.8
+ PruMerge+	57.7	51.2	52.6	62.0	72.1	48.1	21.2	40.5	50.7
+ DART	58.9	54.8	52.2	67.6	69.4	47.0	40.2	39.0	53.6
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 88.89%)*								
+ FastV	82.5	90.3	90.8	94.0	88.7	73.0	41.3	61.7	77.8
+ VisionZip	83.4	89.0	88.1	92.2	92.3	73.0	36.4	74.4	78.6
+ PruMerge+	85.8	87.2	86.4	91.9	94.2	71.6	33.0	73.8	78.0
+ DART	81.2	87.7	86.9	91.7	90.9	70.0	63.2	57.6	78.6
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 88.89%)*								
+ FastV	44.5	39.2	44.1	59.4	46.8	31.0	17.8	28.4	38.9
+ VisionZip	49.4	33.2	44.4	48.1	70.0	30.3	22.0	49.7	43.4
+ PruMerge+	50.4	36.9	38.4	42.9	71.5	28.8	18.1	43.5	41.3
+ DART	47.5	40.5	40.9	49.6	57.7	35.4	31.5	27.3	41.3
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 93.75%)*								
+ FastV	81.4	85.7	86.6	91.5	88.1	74.5	33.0	74.8	77.0
+ VisionZip	79.0	81.9	82.2	88.4	89.4	69.8	25.2	71.3	73.4
+ PruMerge+	76.7	76.9	76.1	87.8	87.6	65.5	21.8	68.9	70.2
+ DART	78.8	81.8	80.4	88.9	88.5	61.8	57.4	67.1	75.6
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 93.75%)*								
+ FastV	35.7	31.9	35.3	48.8	37.4	22.8	13.3	14.8	30.0
+ VisionZip	41.0	34.5	33.3	43.5	66.3	24.3	14.0	26.1	35.4
+ PruMerge+	43.0	29.6	34.1	43.0	67.7	25.5	12.6	29.4	35.6
+ DART	41.9	33.8	38.4	46.9	57.0	26.2	25.6	14.5	35.5
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

downsampling method is sufficient to deal with the visual token compression task.

To validate this hypothesis, we design a data-centric analysis using downsampling as a discriminator. We first drop out the samples answered incorrectly at the original resolution, which we consider are too hard for the original models to understand, not to mention the compressed models. Then, for a given compression ratio, we classify each sample in a benchmark into one of two groups based on the performance of the downsampling method: ① Difficult Samples (Group A): Samples that are answered incorrectly by the downsampling method. ② Simple Samples (Group B): Samples that are answered correctly by the downsampling method. We then evaluate all compression methods on these two groups separately to assess whether the sophisticated methods demonstrate their expected superiority on the "difficult" samples where image downsampling fails. Results from Table 2 and Figure 2 strongly confirm our hypothesis, followed by two key conclusions as follows.

① **Significant performance gap between groups:** Across all benchmarks and compression methods, the accuracy on "simple" samples (Group B) is dramatically higher than on "difficult" samples (Group A). For instance, on GQA at 75% compression, the accuracy of all methods on simple samples is above 87.6%, while on difficult samples, it drops to a maximum of 59.3% (VisionZip). This stark contrast is common in Table 2, demonstrating that the two groups represent essentially different levels of visual comprehension difficulty. The existence of this gap validates our core hypothesis that *the benchmark comprises a mixture of simple and difficult samples. In other words, the current benchmarks are noisy for evaluating the visual token compression methods.* Moreover, the significant gap also proves that *downsampling can serve as a clever filter to distinguish between "simple" and "difficult" samples, which can be the key to denoise the current benchmarks.*

② **Ideal reference points brought by downsampling:** The 0%/100% dichotomy created by image downsampling provides ideal reference points for evaluation. In Group B, where downsampling achieves 100% accuracy, advanced methods show comparable but not superior performance (e.g., 87.6-91.9% on GQA at 75% compression), confirming that their sophisticated approaches offer no advantage for simple samples. In Group A, where downsampling fails completely (0% accuracy), advanced methods demonstrate their true value by significantly exceeding this baseline. For instance, DART achieves 40.2% on OCRBench and VisionZip reaches 51.2% on ChartQA at 75% compression, proving their ability to preserve crucial details that downsampling loses.

3.5 Summary

In this section, we conduct two comprehensive experiments to further understand the anomalous phenomenon: image downsampling exceeds other sophisticated methods under some settings. The first experiment validates the universality of this anomalous phenomenon and introduces our basic hypothesis: Some data in the existing benchmarks is overly simplistic, leading to the unreasonable phenomenon that even the simplest downsampling method is sufficient to deal with the visual token compression task. Furthermore, the second experiment further validates this hypothesis and proves that the current benchmarks are noisy for evaluating the visual token compression methods. Moreover, the second experiment simultaneously demonstrates that downsampling can serve as a clever filter to distinguish between "simple" and "difficult" samples, which can be the key to denoise the current benchmarks.

4 Evaluation Framework

4.1 Framework Construction

To address the simplicity bias and denoise existing benchmarks for the visual token compression task, we propose the VTC-Bench (Visual Token Compression Benchmark) framework, a novel framework specifically designed for the fair and effective evaluation of visual token compression methods. The construction is based on the key insight—validated in Section 3.4—that "downsampling can serve as a clever filter to distinguish between 'simple' and 'difficult' samples". We leverage this idea to construct a challenging benchmark comprising predominantly "difficult" samples that require fine-grained visual understanding. This process, summarized in Figure 3, does not create new data but rather applies a rigorous filtering mechanism to existing benchmarks to identify a challenging evaluation and noise-free set. The pipeline consists of three critical steps executed for each candidate sample and dynamically adapts to different compression ratios:

Step 1: Inference & Compression. Given a sample and a target token compression ratio, we run two inference pipelines: ① a downsampling baseline (the filter) including one model that applies the equivalent ratio from Eq. 1 for a fair comparison and another original model without downsampling, implemented with Qwen2-VL which has a similar number of parameter with the target MLLM; and ② advanced visual token compression methods (e.g., FastV, VisionZip, DART) evaluated directly on the target MLLM. This step both establishes a fair basis for assessing compression approaches and provides signals for subsequent sample filtering.

Step 2: Grouping: We first drop out the samples that are incorrectly answered by the original Qwen2-VL. Then, we use the performance of the downsampling method as a binary discriminator to categorize the sample into two groups: ① Group A: Samples considered as "difficult", which are incorrectly answered by the downsampling method. ② Group B: Samples considered as "simple", which are correctly answered by the downsampling method. This step effectively tags each sample with the labels of "simple" or "difficult", filtering the existing benchmarks and removing noisy data that is not applicable for evaluating the visual token compression methods.

Step 3: Result Aggregation: Based on the classification in Step 2 and the inference results of visual token compression methods in Step 1, we perform a statistical analysis on the accuracy of the "difficult" samples in the methods to be evaluated. Thus, an indicator that can truly reflect the visual compression methods fairly can be obtained.

In summary, we develop VTC-Bench, a simple but effective framework for evaluating visual token compression methods. The building pipeline of VTC-Bench is shown in Figure 4. Importantly, the VTC-Bench framework can be applied easily to any existing benchmark, transforming it into a more effective benchmark for evaluating visual token com-

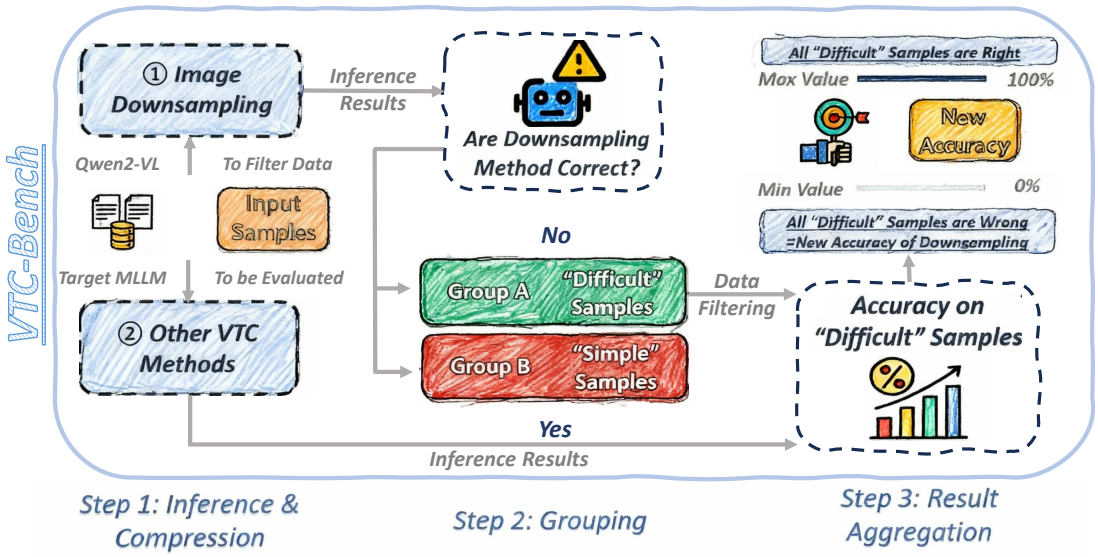

Figure 3: The VTC-Bench is a simple but effective framework that can transform any existing benchmarks to a subset that can fairly evaluate VTC (Visual Token Compression) methods. The samples that are answered correctly by the original Qwen2-VL model without downsampling form the input samples. More details in Sec. 4.1.

Table 3: VTC-Bench results on Qwen2-VL-7B.

Method	GQA	MMB	MMB^{CN}	MME	POPE	MMStar	OCRBench	ChartQA	Average
Qwen-VL-7B	*Token Reduction (↓ 75.00%)*								
+ FastV	57.8	45.2	56.5	78.9	65.4	41.0	29.1	35.0	51.1
+ VisionZip	59.3	42.4	42.2	54.9	72.5	45.9	29.6	51.2	49.8
+ PruMerge+	57.7	51.2	52.6	62.0	72.1	48.1	21.2	40.5	50.7
+ DART	58.9	54.8	52.2	67.6	69.4	47.0	40.2	39.0	53.6
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen-VL-7B	*Token Reduction (↓ 88.89%)*								
+ FastV	44.5	39.2	44.1	59.4	46.8	31.0	17.8	28.4	38.9
+ VisionZip	49.4	33.2	44.4	48.1	70.0	30.3	22.0	49.7	43.4
+ PruMerge+	50.4	36.9	38.4	42.9	71.5	28.8	18.1	43.5	41.3
+ DART	47.5	40.5	40.9	49.6	57.7	35.4	31.5	27.3	41.3
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen-VL-7B	*Token Reduction (↓ 93.75%)*								
+ FastV	35.7	31.9	35.3	48.8	37.4	22.8	13.3	14.8	30.0
+ VisionZip	41.0	34.5	33.3	43.5	66.3	24.3	14.0	26.1	35.4
+ PruMerge+	43.0	29.6	34.1	43.0	67.7	25.5	12.6	29.4	35.6
+ DART	41.9	33.8	38.4	46.9	57.0	26.2	25.6	14.5	35.5
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen-VL-7B	*Token Reduction (↓ 96.00%)*								
+ FastV	29.6	24.7	33.1	35.6	35.6	21.5	11.0	9.3	25.0
+ VisionZip	38.6	31.2	32.6	37.9	60.0	24.5	11.0	15.1	31.4
+ PruMerge+	38.8	26.0	29.3	37.0	56.4	22.6	9.2	17.0	29.5
+ DART	36.4	33.7	36.4	37.9	53.2	22.3	24.7	10.5	31.9
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen-VL-7B	*Token Reduction (↓ 99.00%)*								
+ FastV	18.3	18.3	21.5	21.5	44.3	15.0	4.2	3.8	18.4
+ VisionZip	23.4	28.8	32.2	28.5	53.6	19.4	3.7	5.5	24.4
+ PruMerge+	20.7	17.8	21.1	22.9	52.6	17.1	2.5	7.1	20.2
+ DART	24.5	26.5	28.1	30.6	41.5	19.2	25.6	4.2	25.0
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

pression methods. Meanwhile, the VTC-Bench framework dynamically and reasonably provides a corresponding benchmark subset for each compression ratio, while offering explainable theoretical upper and lower bounds for the final metrics.

4.2 Evaluation Results & Discussions

We conduct extensive experiments across multiple mainstream MLLMs and benchmarks based on VTC-Bench. We select Qwen2-VL-7B (Wang et al., 2024) and LLaVA-OV-7B (Li et al., 2024a) as

Table 4: VTC-Bench results on LLaVA-OV-7B.

Method	GQA	MMB	MMB^{CN}	POPE	MMStar
LLaVA-OV-7B	*Token Reduction (↓ 75.00%)*				
+ FastV	54.3	70.5	69.1	63.8	48.6
+ VisionZip	59.0	67.7	71.3	80.8	44.8
+ PruMerge+	60.4	74.2	73.5	75.6	48.6
+ Downsample	0.0	0.0	0.0	0.0	0.0
LLaVA-OV-7B	*Token Reduction (↓ 88.89%)*				
+ FastV	45.3	64.6	66.4	39.1	42.4
+ VisionZip	56.6	71.9	71.2	69.6	43.5
+ PruMerge+	57.4	68.8	71.5	76.0	45.8
+ Downsample	0.0	0.0	0.0	0.0	0.0
LLaVA-OV-7B	*Token Reduction (↓ 93.75%)*				
+ FastV	36.7	51.2	53.3	29.7	32.6
+ VisionZip	49.1	64.3	62.4	53.6	36.6
+ PruMerge+	50.2	66.6	65.3	59.9	34.8
+ Downsample	0.0	0.0	0.0	0.0	0.0
LLaVA-OV-7B	*Token Reduction (↓ 96.00%)*				
+ FastV	31.4	37.4	43.1	24.5	28.6
+ VisionZip	42.6	55.4	56.9	45.4	30.8
+ PruMerge+	42.7	57.8	59.6	49.9	31.3
+ Downsample	0.0	0.0	0.0	0.0	0.0
LLaVA-OV-7B	*Token Reduction (↓ 99.00%)*				
+ FastV	25.7	25.8	29.6	39.3	21.9
+ VisionZip	28.3	28.1	32.8	42.1	24.7
+ PruMerge+	25.3	25.5	28.5	40.4	25.2
+ Downsample	0.0	0.0	0.0	0.0	0.0

the base MLLMs and evaluate various visual token compression methods (including FastV, VisionZip, PruMerge+, DART) on a subset of "difficult samples" filtered by VTC-Bench. The experimental results are shown in Table 3 and 4 and Figure 4, followed by several analysis:

Is downsampling all you need? Across many benchmarks, simple image downsampling often beats more advanced compression methods, suggesting that sophisticated approaches are unnecessary. VTC-Bench overturns this impression: when we restrict evaluation to the compression-relevant *difficult samples* (Group A), the trend reverses. The apparent superiority of downsampling largely stems from original benchmarks being saturated with *easy* cases that do not require fine-grained cues. By filtering out such samples, VTC-Bench reveals that for truly challenging instances—those that test visual understanding—advanced compression methods are not only effective but necessary.

What makes an effective benchmark? Simple cross-benchmark comparisons (e.g., "Benchmark A outperforms Benchmark B") only imply that one is harder, without revealing *which* skills drive the difficulty or whether it is relevant to visual token compression. VTC-Bench addresses this by filtering out samples that do not inform compression performance, yielding an analysis set that is explicitly sensitive to token compression. This suggests

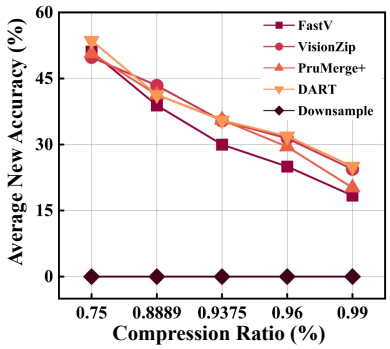

Figure 4: VTC-Bench results on Qwen2-VL-7B.

a design principle for future work: effective benchmarks for visual token compression should deliberately increase the share of compression-relevant hard cases.

Further Expand the Accuracy Gap: VTC-Bench amplifies and clarifies method differences. At 75% compression on ChartQA, the VisionZip–FastV gap widens from 8.8% to 16.2%; at 96% compression on GQA, it grows from 0.3% to 9.0%. These phenomenon effectively indicates that VTC-Bench indeed eliminates data noise unrelated to the visual token compression task, thereby promoting the fairness and effectiveness of the benchmark in the visual token compression task.

5 Conclusion

This paper systematically analyzes the task mismatch problem presented in current MLLM benchmarks when evaluating visual token compression methods. Based on a surprising and counterintuitive finding: simple image downsampling consistently outperforms many advanced compression methods across multiple widely used benchmarks, we conduct a comprehensive empirical study across several advanced visual token compression methods. Thus, two crucial findings are concluded based on the empirical study: ① Current benchmarks are noisy for the visual token compression task. ② Downsampling can serve as a data filter to evaluate the difficulty of samples upon the visual token compression task. Furthermore, we propose VTC-Bench, a new evaluation framework specifically designed to optimize and denoise current existing benchmarks by explicitly distinguishing between "simple" and "difficult" samples through downsampling. Through this work, we hope to not only advance the field of visual token compression but also inspire more discussions within the community on "how to properly evaluate efficient MLLMs."

6 Limitations

① Relying on downsampling as a filter: If downsampling itself performs poorly on certain tasks, it may result in an insufficient number of "difficult samples" being selected. ② Not considering model differences: Different MLLMs have varying sensitivities to image resolution and visual details, which may affect the generalizability of sample grouping.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. 2025. Variation-aware vision token dropping for faster large vision-language models. *arXiv preprint arXiv:2509.01552*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024b. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, and 1 others. 2025. Embodiedeval: Evaluate multimodal llms as embodied agents. *arXiv preprint arXiv:2501.11858*.

Yuqian Fu, Runze Wang, Yanwei Fu, Danda Pani Paudel, Xuanjing Huang, and Luc Van Gool. 2025. Objectrelator: Enabling cross-view object relation understanding in ego-centric and exo-centric videos. *ICCV*.

Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Donglin Wang, Honggang Chen, Qingsen Yan, and Siteng Huang. 2024. Filter, correlate, compress: Training-free token reduction for mllm acceleration. *arXiv preprint arXiv:2411.17686*.

Drew A Hudson and Christopher D Manning. 2019. 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*, pages 6700–6709.

Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. 2025. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4075–4083.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Duo Li, Zuhao Yang, and Shijian Lu. 2025. Todre: Visual token pruning via diversity and task awareness for efficient large vision-language models. *arXiv preprint arXiv:2505.18757*.

Lin Li, Guikun Chen, Hanrong Shi, Jun Xiao, and Long Chen. 2024c. A survey on multimodal benchmarks: In the era of large ai models. *arXiv preprint arXiv:2409.18142*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. 2024a. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*.

Xuyang Liu, Zichen Wen, Shaobo Wang, Junjie Chen, Zhishan Tao, Yubo Wang, Xiangqi Jin, Chang Zou, Yiyu Wang, Chenfei Liao, and 1 others. 2025. Shifting ai efficiency from model-centric to data-centric compression. *arXiv preprint arXiv:2505.19147*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024c. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document

images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Dilxat Muhtar, Zhenshi Li, Feng Gu, Xueliang Zhang, and Pengfeng Xiao. 2024. Lhrs-bot: Empowering remote sensing with vgi-enhanced large multimodal language model. In *European Conference on Computer Vision*, pages 440–457. Springer.

Ruiguang Pei, Weiqing Sun, Zhihui Fu, and Jun Wang. 2025. Greedyprune: Retenting critical visual token set for large vision language models. *arXiv preprint arXiv:2506.13166*.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. 2024. Nuscenes-qa: A multimodal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4542–4550.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. 2024. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Zekun Wang, Minghua Ma, Zexin Wang, Rongchuan Mu, Liping Shan, Ming Liu, and Bing Qin. 2025. Effivlm-bench: A comprehensive benchmark for evaluating training-free acceleration in large vision-language models. *arXiv preprint arXiv:2506.00479*.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. 2025a. Token pruning in multimodal large language models: Are we solving the right problem? *arXiv preprint arXiv:2502.11501*.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025b. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*.

Zichen Wen, Shaobo Wang, Yufa Zhou, Junyuan Zhang, Qintong Zhang, Yifeng Gao, Zhaorun Chen, Bin Wang, Weijia Li, Conghui He, and 1 others. 2025c. Efficient multi-modal large language models via progressive consistency distillation. *arXiv preprint arXiv:2510.00515*.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and 1 others. 2024. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*.

Minhao Xiong, Zichen Wen, Zhuangcheng Gu, Xuyang Liu, Rui Zhang, Hengrui Kang, Jiabing Yang, Junyuan Zhang, Weijia Li, Conghui He, and 1 others. 2025. Prune2drive: A plug-and-play framework for accelerating vision-language models in autonomous driving. *arXiv preprint arXiv:2508.13305*.

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

Senqiao Yang, Junyi Li, Xin Lai, Bei Yu, Hengshuang Zhao, and Jiaya Jia. 2025b. Visionthink: Smart and efficient vision language model via reinforcement learning. *arXiv preprint arXiv:2507.13348*.

Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. 2025c. Efficientvla: Training-free acceleration and compression for vision-language-action models. *arXiv preprint arXiv:2506.10100*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Weihao Yu, Zhengyuan Yang, Lingfeng Ren, Linjie Li, Jianfeng Wang, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Lijuan Wang, and Xinchao Wang. 2024. Mm-vet v2: A challenging benchmark to evaluate large multimodal models for integrated capabilities. *arXiv preprint arXiv:2408.00765*.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2025. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*.

A Experiment Details

In this paper, all the experiments are conducted based on one A800 GPU. For the downsampling method and DART, we apply the official code of DART (Wen et al., 2025b). As to the downsampling method, we resize the image before it

enters the MLLM. As to DART, we control the compression ratio through the parameter "Reduction_Ratio". For VisionZip, PruMerge+, and FastV, we apply the EffiVLM-Bench (Wang et al., 2025), which offers a unified toolkit to evaluate efficient MLLM. As to these three methods, we control the compression ratio through the parameter "Budget". Considering this paper focuses on the evaluation, it is not related to hyperparameter search. All results come from a single run. The code environment includes Python=3.10, torch=2.6.0, torchvision=0.21.0, and torchaudio=2.6.0. We will release all the results, including the output results of each sample and the accuracy results of each benchmark.

B Benchmark Details

B.1 GQA

GQA (Hudson and Manning, 2019) is a large-scale benchmark for visual reasoning and compositional question answering. Based on a strict distribution control, GQA offers 22M valuable reasoning questions.

B.2 MMBench

MMBench (Liu et al., 2024b) is a comprehensive benchmark designed to evaluate the capabilities of MLLMs. It includes 3,217 multiple-choice questions spanning 20 fine-grained dimensions, supporting several languages such as Chinese and English.

B.3 MME

MME (Yin et al., 2024) provides a systematic framework for evaluating the perceptual and cognitive abilities of MLLMs. It encompasses 14 subtasks across the domains of visual perception, text understanding, reasoning, and cross-modal alignment.

B.4 POPE

POPE (Li et al., 2023) is a benchmark designed to evaluate object hallucination in MLLMs. The pipeline of POPE measures hallucination under random, popular, and adversarial sampling strategies.

B.5 MMStar

MMStar (Chen et al., 2024b) is a vision-dependent benchmark for evaluating the reasoning and perception abilities. It has 1500 samples, covering six core abilities with 18 sub-dimensions.

B.6 OCRBench

OCRBench (Liu et al., 2024c) is a comprehensive benchmark for evaluating the OCR capabilities of multimodal large models. The benchmark includes 1,000 manually verified samples from 29 datasets.

B.7 ChartQA

ChartQA (Masry et al., 2022) evaluates visual and logical reasoning over real-world charts. It includes 9.6k human-written and 23.1k automatically generated questions across different kinds of charts.

C Complete VTC-Bench Results

Due to the page limitation, we are unable to offer complete results in the experiment sections. Thus, we provide the evaluation results by group of Qwen2-VL-7B and LLaVA-OV-7B on eight benchmarks here, as shown in Table 5 and 6.

Table 5: Comparison of Advanced Token Compression Methods and Downsampling on Qwen2-VL-7B.

Method	GQA	MMB	MMB^{CN}	MME	POPE	MMStar	OCRBench	ChartQA	Average
Group B	*Token Reduction (↓ 75.00%)*								
+ FastV	87.6	95.9	95.8	96.7	94.8	76.0	57.2	78.1	85.3
+ VisionZip	91.2	93.8	93.6	95.3	96.8	81.4	58.1	87.3	87.2
+ PruneMerge+	91.9	95.1	94.6	95.9	97.5	82.3	46.2	73.6	84.6
+ DART	88.1	94.9	94.6	94.9	94.5	77.7	70.2	69.0	85.5
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 75.00%)*								
+ FastV	57.8	45.2	56.5	78.9	65.4	41.0	29.1	35.0	51.1
+ VisionZip	59.3	42.4	42.2	54.9	72.5	45.9	29.6	51.2	49.8
+ PruneMerge+	57.7	51.2	52.6	62.0	72.1	48.1	21.2	40.5	50.7
+ DART	58.9	54.8	52.2	67.6	69.4	47.0	40.2	39.0	53.6
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 88.89%)*								
+ FastV	82.5	90.3	90.8	94.0	88.7	73.0	41.3	61.7	77.8
+ VisionZip	83.4	89.0	88.1	92.2	92.3	73.0	36.4	74.4	78.6
+ PruneMerge+	85.8	87.2	86.4	91.9	94.2	71.6	33.0	73.8	78.0
+ DART	81.2	87.7	86.9	91.7	90.9	70.0	63.2	57.6	78.6
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 88.89%)*								
+ FastV	44.5	39.2	44.1	59.4	46.8	31.0	17.8	28.4	38.9
+ VisionZip	49.4	33.2	44.4	48.1	70.0	30.3	22.0	49.7	43.4
+ PruneMerge+	50.4	36.9	38.4	42.9	71.5	28.8	18.1	43.5	41.3
+ DART	47.5	40.5	40.9	49.6	57.7	35.4	31.5	27.3	41.3
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 93.75%)*								
+ FastV	81.4	85.7	86.6	91.5	88.1	74.5	33.0	74.8	77.0
+ VisionZip	79.0	81.9	82.2	88.4	89.4	69.8	25.2	71.3	73.4
+ PruneMerge+	76.7	76.9	76.1	87.8	87.6	65.5	21.8	68.9	70.2
+ DART	78.8	81.8	80.4	88.9	88.5	61.8	57.4	67.1	75.6
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 93.75%)*								
+ FastV	35.7	31.9	35.3	48.8	37.4	22.8	13.3	14.8	30.0
+ VisionZip	41.0	34.5	33.3	43.5	66.3	24.3	14.0	26.1	35.4
+ PruneMerge+	43.0	29.6	34.1	43.0	67.7	25.5	12.6	29.4	35.6
+ DART	41.9	33.8	38.4	46.9	57.0	26.2	25.6	14.5	35.5
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 96.00%)*								
+ FastV	79.2	74.5	74.4	90.1	85.8	68.6	27.5	75.3	71.9
+ VisionZip	75.6	80.3	80.2	87.4	86.5	69.6	20.4	69.2	71.2
+ PruneMerge+	72.5	69.3	69.1	83.9	82.2	61.4	18.4	70.3	65.9
+ DART	74.5	77.3	75.1	84.9	84.3	63.5	53.7	71.1	73.1
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 96.00%)*								
+ FastV	29.6	24.7	33.1	35.6	35.6	21.5	11.0	9.3	25.0
+ VisionZip	38.6	31.2	32.6	37.9	60.0	24.5	11.0	15.1	31.4
+ PruneMerge+	38.8	26.0	29.3	37.0	56.4	22.6	9.2	17.0	29.5
+ DART	36.4	33.7	36.4	37.9	53.2	22.3	24.7	10.5	31.9
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 99.00%)*								
+ FastV	75.5	55.6	53.2	75.3	59.3	61.0	19.5	73.3	59.1
+ VisionZip	79.4	78.3	76.9	80.5	70.2	69.4	17.1	70.3	67.8
+ PruneMerge+	73.9	55.9	52.6	73.9	49.7	57.0	13.0	69.5	55.7
+ DART	76.1	63.2	59.0	73.2	67.4	63.7	43.1	70.7	64.6
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 99.00%)*								
+ FastV	18.3	18.3	21.5	21.5	44.3	15.0	4.2	3.8	18.4
+ VisionZip	23.4	28.8	32.2	28.5	53.6	19.4	3.7	5.5	24.4
+ PruneMerge+	20.7	17.8	21.1	22.9	52.6	17.1	2.5	7.1	20.2
+ DART	24.5	26.5	28.1	30.6	41.5	19.2	25.6	4.2	25.0
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 6: Comparison of Advanced Token Compression Methods and Downsampling on LLaVA-ov-7B

Method	GQA	MMB	MMB^{CN}	POPE	MMStar	Average
Group B	*Token Reduction (↓ 75.00%)*					
+ FastV	84.0	93.5	94.7	92.2	73.1	87.5
+ VisionZip	86.2	93.4	94.2	95.6	62.9	86.5
+ PruneMerge+	87.1	93.8	94.0	96.3	62.1	86.7
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 75.00%)*					
+ FastV	54.3	70.5	69.1	63.8	48.6	61.3
+ VisionZip	59.0	67.7	71.3	80.8	44.8	64.7
+ PruneMerge+	60.4	74.2	73.5	75.6	48.6	66.5
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 88.89%)*					
+ FastV	76.1	91.7	92.3	85.2	66.5	82.4
+ VisionZip	82.8	92.5	92.1	93.6	59.1	84.0
+ PruneMerge+	82.9	92.8	93.2	93.8	54.9	83.5
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 88.89%)*					
+ FastV	45.3	64.6	66.4	39.1	42.4	51.6
+ VisionZip	56.6	71.9	71.2	69.6	43.5	62.6
+ PruneMerge+	57.4	68.8	71.5	76.0	45.8	63.9
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 93.75%)*					
+ FastV	73.0	85.8	86.5	81.5	64.3	78.2
+ VisionZip	79.4	91.2	90.9	90.9	54.6	81.4
+ PruneMerge+	78.7	91.1	91.0	91.0	53.6	81.1
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 93.75%)*					
+ FastV	36.7	51.2	53.3	29.7	32.6	40.7
+ VisionZip	49.1	64.3	62.4	53.6	36.6	53.2
+ PruneMerge+	50.2	66.6	65.3	59.9	34.8	55.4
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 96.00%)*					
+ FastV	71.9	77.3	77.9	79.0	57.1	72.6
+ VisionZip	76.3	86.9	86.8	86.6	49.9	77.3
+ PruneMerge+	74.5	84.1	84.2	85.9	49.7	75.7
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 96.00%)*					
+ FastV	31.4	37.4	43.1	24.5	28.6	33.0
+ VisionZip	42.6	55.4	56.9	45.4	30.8	46.2
+ PruneMerge+	42.7	57.8	59.6	49.9	31.3	48.3
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0
Group B	*Token Reduction (↓ 99.00%)*					
+ FastV	63.0	50.6	46.5	59.0	47.4	53.3
+ VisionZip	64.4	54.5	53.8	61.8	36.0	54.1
+ PruneMerge+	60.4	46.9	45.5	56.7	32.6	48.4
+ Downsample	100.0	100.0	100.0	100.0	100.0	100.0
Group A	*Token Reduction (↓ 99.00%)*					
+ FastV	25.7	25.8	29.6	39.3	21.9	28.5
+ VisionZip	28.3	28.1	32.8	42.1	24.7	31.2
+ PruneMerge+	25.3	25.5	28.5	40.4	25.2	29.0
+ Downsample	0.0	0.0	0.0	0.0	0.0	0.0