TABED: Test-Time Adaptive Ensemble Drafting for Robust Speculative Decoding in LVLMs

Anonymous ACL submission

Abstract

Speculative decoding (SD) accelerates the decoding stage by speculating multiple next tokens with a small draft model, which is, in turn, verified by the target model in parallel. Despite its success in LLM inference acceleration, SD largely remains unexplored for Large Vision Language Models (LVLMs), an advanced class of LLMs that can handle multimodal prompts consisting of text and image tokens. To bridge this gap, we first evaluate a comprehensive scenario based on real-world deployments of LVLM SD. We observe that drafting with and without image tokens using a small draft model exhibits scenario-specific performance fluctuations. Motivated by this, we propose Test-time Adaptive Batched Ensemble Drafting, a fully *training-free* yet effective SD method for LVLMs. Our method leverages multiple drafting methods via batch inference. It dynamically weights these drafts based on their deviation from the target model's previous outputs. To further enhance its extensibility at negligible cost, we incorporate alternative drafting strategies, such as image captioning and pooling. Our method achieves an average speedup of 1.8x while maintaining robustness across diverse input scenarios. Since our method relies solely on the draft model without incurring additional costs, it is fully compatible with existing LVLM acceleration techniques and can be seamlessly integrated into them. To ensure reproducibility, we open-source our code and custom-trained draft LVLMs.

1 Introduction

Multimodal Large Language Models (MLLMs) (Yin et al., 2024; Wu et al., 2023; Zhang et al., 2024a) are an advanced class of LLMs (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023) designed to process multiple modalities, such as images, audio, and video, alongside text. In particular, *Large Vision Language Models (LVLMs)* (Chen et al., 2024c), which can handle prompts comprised of *text and images*—also known as Large Multimodal Models (Li et al., 2024b; Jin et al., 2024; Song et al., 2023)—have attracted significant attention due to their unique applications, including multimodal chatbots, visual data analysis, and augmented reality (AR) applications, among others (OpenAI, 2023; Anthropic, 2024; Gemini Team Google: Anil et al., 2023).

As LVLMs are increasingly deployed, reducing their inference time has become a critical issue. In addition to the standard LLM inference steps, LVLMs must (1) preprocess images in the input prompt to obtain *image tokens* (typically several hundred tokens per image) (Radford et al., 2021; Liu et al., 2023, 2024a) and (2) process both text and image prompts, resulting in considerably higher inference latency. Therefore, accelerating LVLM inference is of substantial practical importance.

Recently, methods like token pruning, layer skipping, and key-value cache compression have been proposed to accelerate LVLM inference (Shang et al., 2024; Chen et al., 2024b; Lin et al., 2024; Liu et al., 2024d; Wan et al., 2024; McKinzie et al., 2024). While effective, these approximation techniques cannot fully preserve the original LVLM's output distribution. Moreover, they primarily reduce prompt processing time (prefilling stage) but have limited impact on response generation time (decoding stage), making them less effective for long outputs.

Speculative Decoding (SD) (Leviathan et al., 2023; Chen et al., 2023) is a large language model (LLM) inference acceleration technique that fully preserves the output distribution. SD first swiftly *speculates* a specified number of draft tokens and then uses the original target model to *verify* these tokens simultaneously. For LLM inference, SD has proven highly effective by further developing both the drafting and verification stages (Miao et al., 2023; Li et al., 2024c; Cai et al., 2024). These methods often leverage additional training, either for a separate lightweight draft model or an added component within the target model, to align with their specific approach (Xia et al., 2024; Zhou et al., 2024). In this context, compatibly combining SD methods for each component has become increasingly important for maximizing overall speedup.

Unlike LLM inference, SD for LVLMs has been far less explored, with the only notable exception by Gagrani et al. (2024). They successfully accelerated LVLM inference via SD for the first time using text-only drafts (i.e., the draft model relied solely on the text tokens and ignored the image tokens)—an observation that might seem both intriguing and counterintuitive. Unfortunately, the authors did not provide a detailed analysis of this finding, underscoring the need for a more in-depth study of SD for LVLMs.

Motivated by this, we first aim to analyze SD by reproducing Gagrani et al. (2024) on a larger scale. To account for real-world deployments, we evaluate diverse scenarios, including benchmark and out-of-distribution (OOD) multimodal datasets, and further experiment with multi-turn interactions, exploring various types of subsequent requests—those with or without images, and those either dependent on or independent of prior responses. Our results show that, with a sufficiently small draft model capable of accelerating LVLM inference through SD, multimodal and text-only drafting approaches each have their own strengths and weaknesses, depending on the input scenario and target response type. Thus, selecting between multimodal and text-only drafting is a nontrivial challenge. This raises the question: can we design a drafting approach that combines the strengths of both methods to achieve robust performance across all input scenarios?

Motivated by these observations, we present TABED, which leverages (1) multiple drafts simultaneously through batch inference and (2) dynamically weights them according to how much each differs from the ground truth labels of the target model after verification. This takes advantage of the fact that batch inference with small models can be scaled without incurring additional costs. The test-time adaptation process, based on inference outputs during SD, is a fully *training-free* approach and does not require any additional parameters.

By doing so, TABED achieves an average speedup of 1.8x and also robust across diverse input scenarios, unlike single-drafting methods, which often exhibit scenario-specific performance fluctuations. More specifically, compared to single drafting methods such as multimodal drafting and text-only drafting, our approach achieves an average performance improvement of over 5%. This is particularly significant because the gain is achieved solely through through batch inference and test-time ensemble learning of distributional outputs from single drafting methods, without any additional training.

In addition, as our approach generalizes to more than three draft methods, we explore unexplored drafting methods, such as image captioning and pooling. We find that these methods also offer complementary advantages, further enhancing the robustness of our framework when integrated.

As the enhanced speed and robustness are achieved using only the draft model itself without incurring additional costs, our approach is fully compatible with and can be seamlessly integrated into existing LVLM acceleration techniques, including those targeting the *Prefilling Stage* or other types of SD. To establish a solid foundation in the emerging field of LVLM SD and ensure full reproducibility of our work, we open-source our code and the custom-trained draft LVLMs at (See Appendix E for more details).

2 Related Work

2.1 Large Vision Language Models

LVLMs Frontier proprietary LVLMs (OpenAI, 2023; Anthropic, 2024; Gemini Team Google: Anil et al., 2023) demonstrate state-of-the-art performance across multimodalities beyond just text. Meanwhile, open-source models like the LLaVA series (Liu et al., 2023, 2024a; Li et al., 2024b,a) and LLaMA 3.2 (Dubey et al., 2024) are also rapidly advancing. While various methods exist for embedding image inputs (Yin et al., 2024; Jin et al., 2024), one of the most prominent approaches, LLaVA, employs an off-the-shelf vision encoder (Radford et al., 2021; Zhai et al., 2023) and a trainable projector to convert each image into several hundred visual context tokens of an LLM.

Approximate Inference To address the inefficiency of handling visual tokens from images, several approaches have been proposed based on a common finding: only a sparse subset of the hundreds of visual tokens is important, allowing for reduced computational cost with minimal information loss. Shang et al. (2024); Chen et al. (2024b);

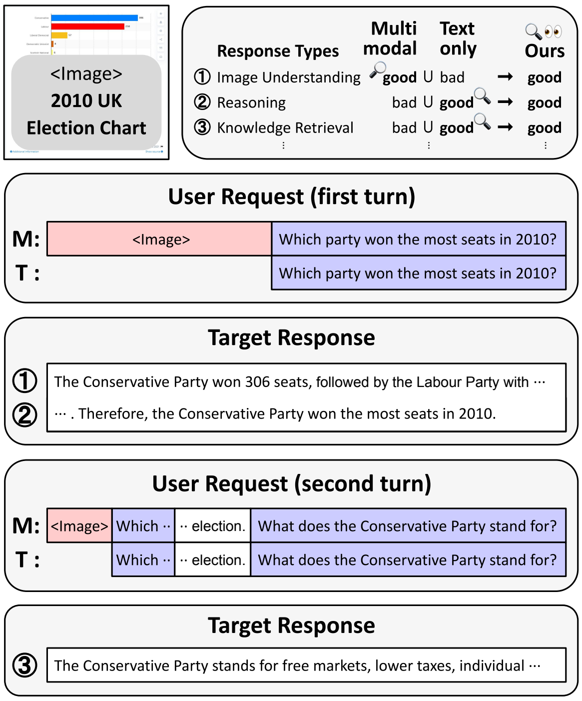


Figure 1: **Overview of TABED** Considering scenarios for real-world SD deployments, existing single drafting methods (M, T) with small draft models exhibit performance fluctuations, as target model responses require different drafting types in both intra-response (① to ②) and inter-response (③) cases. In a fully training-free manner, **TABED** addresses this by tracking changes through deviations from the target model's previous outputs, using multiple drafts obtained via batch inference. By doing so, we achieves both improved speed and robustness across diverse input scenarios. See Fig. 2 for method details.

Lin et al. (2024) dynamically prune significant visual tokens based on attention sparsity. Further focusing on reducing redundant key-value caches, Liu et al. (2024d); Wan et al. (2024) retain key-value vectors by merging or discarding less critical caches during inference. However, from a latency perspective, these approaches primarily benefit the prefilling stage while providing limited advantages for the decoding stage.

2.2 Speculative Decoding

SD for LLMs SD accelerates LLM inference using a small draft model while preserving the target model's output distribution (Leviathan et al., 2023; Chen et al., 2023). To improve the drafting phase, various efforts have been made, including generating multiple draft candidates (Miao et al., 2023; Sun et al., 2024b; Yang et al., 2024), and fine-tuning the draft model with knowledge distillation (Zhou et al., 2024). Some studies address cases with exceptionally long prefill lengths (e.g., 100k), which significantly affect decoding efficiency (Sun et al., 2024a; Chen et al., 2024a).

SD for LVLMs Gagrani et al. (2024) is the only prior work that studied SD for LVLMs.[1] They introduced text-only drafting and claiming its performance is comparable to multimodal drafting. However, their benchmark results and detailed analysis of each drafting were limited, and they did not address how to best use multimodal information for improved drafting. Furthermore, whether or not one can effectively use multiple drafting methods remains unclear.

2.3 Test-time Adaptation

Test-time adaptation (Liang et al., 2025) aims to enhance model robustness by adapting to distribution shifts that occur during inference. Since test-time adaptation methods are designed for real-world testing scenarios, they assume that only input data can be utilized without access to the corresponding ground truth labels (Wang et al., 2021, 2022).

3 Preliminaries

3.1 Theoretical Latency of Transformers

Following (Chen et al., 2024a), for a given batch size B and a sequence length S, let $T(B, S, 1)$ denote the time to decode a single token and $T(B, S, \gamma)$ the time to verify γ tokens in parallel. Under moderate S (*e.g.*, $S \leq$ 3k) and sufficiently small B (*e.g.*, $B \leq 4$) and γ (*e.g.*, $\gamma \leq 10$), the decoding phase displays the following observations (Chen et al., 2024a; Fu, 2024), where $\Delta T = T_{\text{max}} - T_{\text{min}}$ denotes the maximum time difference across the varying parameter in each remark:

Remark 1. *For given B and S, regardless of γ, $T(B, S, \gamma)$ remains approximately constant* (e.g., *$|\Delta T/T| < 0.05$).*

Remark 2. *For a given B, regardless of S, $T(B, S, 1)$ remains approximately constant* (e.g., *$|\Delta T/T| < 0.05$).*

Remark 3. *For a given S, regardless of B, $T(B, S, 1)$ remains approximately constant* (e.g., *$|\Delta T/T| < 0.05$).*

Note that the magnitude of the relative difference $|\Delta T/T|$ depends on various factors, such as model

[1] Jang et al. (2024) and Teng et al. (2024) propose using SD for accelerating text-to-image generative models, which are different from LVLMs.

architecture, model size, and hardware specifications. We empirically demonstrate Remarks 1 to 3 in Appendix D.

3.2 Speculative Decoding

Algorithm Following (Leviathan et al., 2023; Zhou et al., 2024), let M_p be the target model whose inference we aim to accelerate, and let M_q be the draft model for the same task. For a given prefix x, generated sequence $y_{<t}$, chunk length γ, and $n = 0, \ldots, \gamma - 1$, the following steps are repeated until either an <EOS> token is accepted or the maximum sequence length is reached:

1. The *Drafting Phase*, where M_q sequentially generates γ draft tokens from $q(y_{t+n}|x, y_{<t+n})$.

2. The *Verification Phase*, where M_p reviews these draft tokens in parallel, comparing them to $p(y_{t+n}|x, y_{<t+n})$.

3. For sampling, each token y_{t+n} is sequentially accepted with probability $\min\left(1, \frac{p(y_{t+n}|x, y_{<t+n})}{q(y_{t+n}|x, y_{<t+n})}\right)$. If any token is rejected before the end of the block, subsequent tokens are discarded, and the rejected token is resampled from the adjusted distribution $\mathrm{norm}(\max(0, p(y) - q(y)))$.[2]

Block Efficiency and Wall-clock Time Improvement Given input, the *block efficiency* $\tau_{p,q}(\gamma)$ is defined as the expected number of accepted tokens per block. Let $T_p(B, S, 1)$ and $T_q(B, S, 1)$ denote the time required for M_p and M_q, to decode a single token, and $T_p(B, S, \gamma)$ denote the time required for M_p to verify γ tokens in parallel. For brevity, we use the simplified notations T_p, T_q, and $T_p(\gamma)$, omitting B and S. The required time per block in SD, denoted as T_{SD}, can be approximated as $T_{\text{SD}} = \gamma \cdot T_q + T_p(\gamma) \approx \gamma \cdot T_q + T_p$ by Remark 1. The *token rate* is defined as the number of tokens generated per unit time. SD's wall-clock time improvement can be expressed as the *token rate ratio:*

$$\frac{\text{Token rate (SD)}}{\text{Token rate (target)}} = \frac{\tau_{p,q}(\gamma)/T_{\text{SD}}}{1/T_p} \approx \frac{\tau_{p,q}(\gamma)}{\gamma \cdot \frac{T_q}{T_p} + 1} \tag{1}$$

Both the block efficiency $\tau_{p,q}(\gamma)$ and the *draft-to-target latency ratio* $\frac{T_q}{T_p}$ are determined by the choice of M_q, assuming M_p is fixed. Remarks 2 and 3 imply the following:

Remark 4. *For a given γ, regardless of B and S, $T_{SD}/T_p = \gamma \cdot \frac{T_q}{T_p} + 1$ remains nearly identical.* (e.g.*if we assume $T_q/T_p = 0.05$ and $\gamma = 5$, $|\Delta T_{SD}/T_{SD}| < 0.01$).*

Remark 4 shows that the wall-clock time improvement in Eq. (1) becomes proportional solely to the value of $\tau_{p,q}(\gamma)$, since its denominator $\gamma \cdot \frac{T_q}{T_p} + 1$ is constant. Moreover, when measuring the actual wall-clock time, precise performance comparison becomes challenging due to potential noise from various factors such as hardware variations. Therefore, we utilize block efficiency $\tau_{p,q}(\gamma)$ to accurately evaluate the performance of speculative decoding.

4 Benchmarking SD for LVLMs

In this section, we systematically study speculative decoding for LVLMs, evaluating the performance of multimodal and text-only drafting across various benchmark datasets.

4.1 Experiment Settings

Target and Draft Models We employ LLaVA-1.5 7B (Liu et al., 2024a), one of the most widely adopted public LVLMs, as our target model. To effectively accelerate the target model using SD, the *draft-to-target latency ratio*—represented by $\frac{T_q}{T_p}$ in Eq. (1)—is crucial. Since no sufficiently small LLaVA models are publicly available, we train our draft model using small public LLaMA variants, with 68M (Miao et al., 2023) as our primary model size, along with 160M (Miao et al., 2023).[3]

For comprehensive benchmarking, we developed three distinct variants of the draft model using different training strategies. First, we trained a draft model following the LLaVA-1.5 training recipe to align with the target model. Second, we trained another draft model using the LLaVA-OV (Li et al., 2024a) training recipe, which specializes in multi-image processing. Finally, we used the base LLaMA model without any LVLM training as the third draft model variant. See Appendix E.1 for more details about draft model training.

To summarize, we conducted benchmarks on draft models across i) model sizes: 68M, 160M, and ii) model types: LLaVA-1.5, LLaVA-OV. We particularly focused on the results of LLaVA-1.5

[2]Whenever the prefix $(x, y_{<t})$ is clear from the context, we'll use $p(y)$ and $q(y)$ to denote $p(y_t|x, y_{<t})$ and $q(y_t|x, y_{<t})$, respectively.

[3]LLaVA-OneVision 0.5B (Li et al., 2024a) is not suitable as a draft model since its latency ratio T_q/T_p to the 7B model of the same series exceed 0.5 (QwenTeam).

Draft Model			Benchmark Datasets (First Turn)								OOD Datasets	
Type	Size	Method	LLaVA-W	DocVQA	POPE	MMVet	IEdit	MB	Spot	Avg.	PSV	VIST
LLaVA-1.5	68M	M	2.28	2.15	2.56	2.21	2.19	1.96	2.34	2.24	1.19	1.16
		T	2.19	2.08	2.31	2.16	2.23	2.34	2.27	2.23	2.05	2.05
		MT	2.25	2.15	2.47	2.21	2.31	2.37	2.4	2.31	1.94	1.91
		MT*	2.26	2.16	2.52	2.21	2.29	2.39	2.36	2.31	2.02	2.04
Draft Model			Benchmark Datasets (Second Turn)								NLP Datasets	
Type	Size	Method	LLaVA-W	DocVQA	POPE	MMVet	IEdit	MB	Spot	Avg.	NQ	GSM8K
LLaVA-1.5	68M	M	2.1	1.96	2.78	2.18	1.61	1.53	1.83	2.00	1.98	2.25
		T	2.32	2.23	2.91	2.56	1.87	2.01	2.08	2.28	2.03	2.30
		MT	2.29	2.24	2.93	2.54	1.81	1.91	2.02	2.25	2.02	2.28
		MT*	2.29	2.23	2.93	2.56	1.85	1.99	2.05	2.27	2.03	2.29

Table 1: Block efficiency results for multimodal drafting (M) and text-only drafting (T) are presented. **MT** uses training-free ensemble learning via batched inference, while **MT*** dynamically weights these using test-time adaptation. Across all combinations of turn-taking, image inclusion, contextual relatedness, and datasets (benchmark and OOD), **MT*** consistently matches or outperforms the best single draftings, while M and T vary by scenario. These results are achieved using draft models only, with no additional costs during training or inference. See Appendix A for details on various model sizes and types.

68M, which has the smallest model size and a training distribution well-aligned with the target model. We separately evaluated the model on multimodal tasks and observed that the trained model can perceive multimodality (see Appendix E.2).

Input Scenarios Given their use in real-world deployments, SD systems must perform reliably across diverse input scenarios, ensuring both consistent speedup and stability. To evaluate this, we curated seven benchmark datasets to address the lack of established benchmarks for LVLM SD systems. These datasets cover both single-image tasks (Liu et al., 2023; Mathew et al., 2021; Li et al., 2023b; Yu et al., 2023) and multi-image scenarios (Tan et al., 2019; Zhang et al., 2024b; Jhamtani and Berg-Kirkpatrick, 2018). We also included two datasets with multi-image ($n = 5$) inputs (Li et al., 2019; Huang et al., 2016) as notable Out-of-Distribution (OOD) cases, where the system must handle unexpected, significantly different queries while maintaining consistent speedup.

We further evaluated multi-turn scenarios using benchmark datasets that incorporate various types of follow-up queries, including those dependent on prior responses (e.g., follow-up requests with images from the same dataset or text-only tasks from multi-turn benchmarks for LVLMs (Liu et al., 2024b)) and distinct text-only reasoning tasks (Kwiatkowski et al., 2019; Cobbe et al., 2021). For all scenarios, the reliability of the target model's responses was evaluated (see Appendices C and G for details and qualitative examples).

Drafting Methods: Multimodal and Text-only The multimodal drafting follows the standard LVLM process and accepts both images and text. In contrast, text-only drafting, which was first explored in (Gagrani et al., 2024), uses only textual data as input for the draft model and follows the standard LLM process. We set $\gamma = 5$ and perform greedy decoding with a maximum of 128 new tokens for all experiments in the paper, including this section.

4.2 Experimental Results

Table 1 presents the block efficiency results for multimodal (M) and text-only (T) drafting. Regardless of model size or multi-image awareness capabilities, multimodal drafting achieves higher block efficiency than text-only drafting in single-turn cases across most benchmark datasets. However, text-only drafting shows comparable overall performance and even outperforms multimodal drafting in subsequent turns and OOD cases.

Summary: No One-Size-Fits-All Drafting Both text-only and multimodal drafting exhibit scenario-specific performance fluctuations when using draft models small enough for SD in LVLM inference, with neither method consistently outperforming the other across various input scenarios. It's hard to know in advance which one is better before execution, and even if known, addressing inconsistency with a single drafting method is difficult.

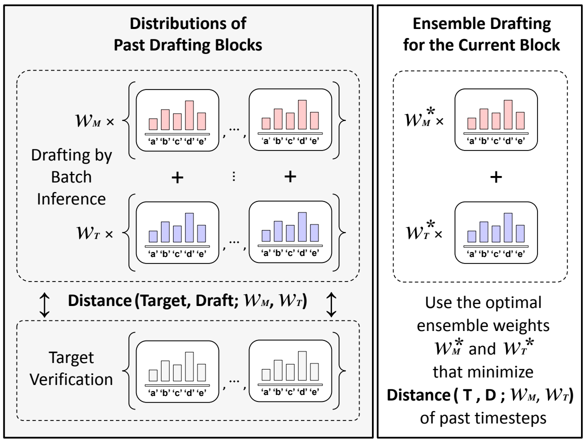


Figure 2: TABED first employs multiple drafting methods via batch inference. It then predicts the optimal weight, w^*, to dynamically adjust the weights of these drafts based on the deviation of previous drafting blocks from the target model's prior outputs. This approach is fully training-free and easily extendable to support additional drafting strategies (Section 6) with negligible cost. See Algorithm 1 for further details.

Algorithm 1 TABED

Parameter: M_q, Prefix $X = [x^{(1)}, ..., x^{(m)}]$, $W =$ weight list of $W_j (= [w^{(1),j}, ..., w^{(m),j}])$

▷ W-AVG stands for Weighted Average

▷ D stands for Distance

Input: Batch $b_{<t} := [(x^{(1)}, y_{<t}), ..., (x^{(m)}, y_{<t})]$

Output: γ draft tokens $y_t, ..., y_{t+\gamma-1}$ and ensembled probabilities $q_t, ..., q_{t+\gamma-1}$

```
procedure TABED(b_{<t}; γ, W)
    q^j = [q(·|x, y_{<1}; W^j), ..., q(·|x, y_{<t}; W^j)]
    p_{t'} = [p(y_{<1}), ..., p(y_{<t})]]
    W_t^* ← arg min_j D(p_{t'}, q^j; W^j)
    for i ← 0 to γ − 1 do
        [q_{t+i}^{(1)}, ..., q_{t+i}^{(m)}] ← M_q(b_{<t+i})
        q_{t+i} ← W-AVG([q_{t+i}^{(1)}, ..., q_{t+i}^{(m)}]; W_t^*)
        y_{t+i} ← SAMPLE(q_{t+i})
    end for
    return [y_t, ..., y_{t+γ−1}], [q_t, ..., q_{t+γ−1}]
end procedure
```

5 Test-time Adaptive Batched Ensemble Drafting

In this section, we propose Test-time Adaptive Batched Ensemble Drafting (TABED), a *training-free* drafting method designed to handle diverse input scenarios for LVLMs which leverages (1) multiple draftings simultaneously through batch inference and (2) the dynamic combination of multiple drafting inferences by adapting the weight for each. This process can be seamlessly integrated with existing LVLM acceleration techniques and other forms of SD. Fig. 2 illustrates the TABED framework.

5.1 Proposed Method

Ensemble learning via Batched Inference Ensemble learning (Dietterich, 2000; Ganaie et al., 2022) is a powerful technique that combines multiple predictions to improve performance. This is particularly beneficial for small models with limited capacity and high bias, as ensembling helps reduce both bias and variance in predictions (Zhou, 2012). However, it often requires additional parameters or models to generate multiple predictions.

For a training-free and robust drafting method, we use ensembling based on multiple predictions generated through batched inference tailored for LVLMs. At each decoding timestep, all m draftings share the parameters of the draft model M_q, and their distributions are ensembled to sample the next token in the draft candidate (see Algorithm 1 for details). We use a weighted averaging ensemble method, sampling a token from the ensembled distribution to continue drafting. This approach is extensible, as it incurs no additional costs during either the training or inference stages—requiring no further training and causing no inference slowdown (Eq. equation 1, Remarks 3 and 4).

Test-time Adaptive Ensemble Weights While using fixed equal weights in ensemble learning can be effective, performance can be further improved by dynamically adjusting these weights W. This allows for a more effective combination of probabilities $[q_t^{(1)}, \ldots, q_t^{(m)}]$, making the ensembled distribution $q(\cdot|x, y_{<t}; W)$ more closely align with the target distribution $p(\cdot|x, y_{<t})$. For instance, the weight assigned in multimodal drafting may vary based on the importance of visual context for a specific sample x and timestep t. To achieve this, we explore test-time adaptation for dynamic weights.

Unlike previous works on test-time adaptation (Wang et al., 2021, 2022), in the case of SD, after completing the verification of all steps prior to t, we gain access to both hard labels (i.e., $y_{<t}$) and soft labels (i.e., $p(y_{t'})$ for $t' < t$) for these earlier steps. This information can be leveraged when drafting restarts from step t, allowing for the dynamic adjustment of each drafting method's influence based on its performance.

To achieve this, we explore dynamic weighting rules that leverage both types of labels. Specifically, at the beginning of a new drafting block, we first sample n weight lists $W^j = [w^{(1),j}, \dots, w^{(m),j}]$ where $j = 1, \dots, n$ (Algorithm 1). Using each W^j, we can generate n trajectories of length $t-1$, where each trajectory consists of ensembled draft probabilities $q^j_{t'} = q(\cdot|x, y_{<t'}; W^j)$. From these trajectories, we sample token trajectories and select the best W^j that maximizes the number of matches with the hard labels $y_{<t}$. We can further use soft labels $p(y_{t'})$ in a complementary manner to select the best W^j that minimizes the accumulated error e^j_t over the previous steps t'.

$$e^j_t = \sum_{t'<t} D_{\mathrm{KL}}\big(p(\cdot \mid x, y_{<t'}) \parallel q^j_{t'}(\cdot \mid x, y_{<t'})\big) \tag{2}$$

where D_{KL} is the KL divergence between p and $q^j_{t'}$ at each of the previous steps t'. This weight list W^*_t is used throughout the current drafting block of γ tokens (i.e., from timestep t to $t+\gamma-1$). Drafting methods with higher weights indicate closer alignment to the target model. This approach dynamically adjusts their influence during the drafting process.

5.2 Experimental Results

Table 1 presents the block efficiency results of TABED across various input scenarios. MT*, which utilizes two existing single draftings (M and T), demonstrates comparable or superior performance to the best single draftings (M or T) across all combinations of turn-taking, image inclusion, relatedness to prior context, and datasets (both benchmark and OOD). Even when MT* achieves the second-best results, its performance remains very close to the best, unlike the wider performance gap observed with other methods. These consistent average performance improvements underscore its ability to effectively distinguish between strong and weak drafting methods during test time and assign optimal weights by dynamically adapting as more history accumulates. This is further validated through variable control in the comparison between MT* and MT, a simpler ensemble drafting method with a fixed 1:1 weight ratio.

Draft Model			Benchmark		OOD
Type	Size	Method	First	Second	Avg.
LLaVA-1.5	68M	M	2.24	2.00	1.18
		T	2.23	2.28	2.05
		C	2.29	2.30	2.09
		P	2.23	2.25	2.08
		MTCP	2.32	2.31	2.12

Table 2: Block efficiency results for single draftings (multimodal (M), text-only (T), caption (C), and pooled-multimodal (P)) are presented. Notably, MTCP applies batched inference to draftings without adding training or inference costs. Using only batched inference, MTCP consistently achieves the highest block efficiency across both the first and second turns in all benchmark and OOD datasets compared to all single draftings. Block efficiency is averaged across datasets in each category.

6 Extensible Framework: Exploring Drafting Candidates for LVLMs

In LVLM SD setting: (1) image tokens have relatively sparse importance and receive meaningful attention only in certain layers compared to text tokens (Shang et al., 2024; Chen et al., 2024b) and (2) the image in the input can be manipulated in various ways and appended to the text, unlike in the LLM SD setting. Building on these observations, we explore and benchmark two alternative drafting schemes: *pooled multimodal drafting* and *caption drafting*, which provide visual information to the draft model through different mechanisms. These drafting schemes can be seamlessly integrated into our proposed method without incurring additional costs during training or inference.

6.1 Pooled Multimodal Drafting

We apply average pooling during inference to compress image information while preserving the 2D spatial structure just before the projector transforms it into the text embedding space. Specifically, we use a 2×2 pooling kernel, reducing the number of visual tokens from 576 to 144 as our default configuration.

6.2 Caption Drafting

To condense sparsely important image tokens into textual descriptions, we employ a lightweight image captioning model (Li et al., 2022, 2023a; Xiao et al., 2024). The captioning model runs only once, in parallel with the target model's prefilling stage, so it is effectively hidden within the prefill phase of the target model. Any minor delays, such as those caused by hardware variations, are amortized across the entire decoding process (Fig. 7). We use Florence as the default captioning model (see Appendix B for further details).

6.3 Experimental Results

Leveraging the extensibility of our approach, we incorporate the alternative drafting candidates pooled-multimodal drafting (P) and caption drafting (C) into the framework into M and T, and evaluating their performance on the same scenarios. MTCP further achieves significant performance gains in most scenarios, due to the differently augmented visual information introduced by P and C. As shown in Table 2, P and C retain the strong robustness of text-only drafting while achieving better overall performance on both benchmark and OOD datasets. This indicates that the condensed visual context produced by P and C exhibits a certain degree of image awareness. Notably, with more drafting candidates, MTCP using only batched inference consistently achieves the highest block efficiency compared to M, T, C, and P across both the first and second turns in all benchmark and OOD datasets. Block efficiency is averaged across datasets in each category.

7 Conclusion

In this work, we analyze existing drafting methods for LVLM SD exhibit scenario-specific performance fluctuations across diverse scenarios relevant to real-world deployments. To address this, we propose TABED, a training-free framework that combines multiple drafting methods via batch inference and dynamically adapts their contributions at test time to leverage small draft models in SD settings. Our approach achieves a 1.8x speedup and over 5% performance improvement compared to single drafting methods, with no additional training or parameters. We further enhance TABED by integrating unexplored methods, improving its robustness and versatility across diverse scenarios.

8 Limitations and Future Works

While we focused on a single draft candidate and single verification scheme to understand the fundamentals of LVLM SD, one may use multiple draft candidates (Miao et al., 2023; Yang et al., 2024; Cai et al., 2024). Being orthogonal to these approaches, our method is easily compatible with them and could benefit from such integrations. We believe that our approach is applicable to other MLLMs such as those for audio and text tokens (Fu et al., 2024).

References

Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Jian Chen, Vashisth Tiwari, Ranajoy Sadhukhan, Zhuoming Chen, Jinyuan Shi, Ian En-Hsu Yen, and Beidi Chen. 2024a. MagicDec: Breaking the Latency-Throughput Tradeoff for Long Context Generation with Speculative Decoding. *arXiv preprint arXiv:2408.11049*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024b. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *arXiv preprint arXiv:2403.06764*.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. 2024c. Dress: Instructing large vision-language models to align and interact with humans via natural language feedback. *Preprint*, arXiv:2311.10081.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Xiong Wang, Di Yin, Long Ma, Xiawu Zheng, et al. 2024. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*.

Yao Fu. 2024. Challenges in Deploying Long-Context Transformers: A Theoretical Peak Performance Analysis. *arXiv preprint arXiv:2405.08944*.

Mukul Gagrani, Raghavv Goel, Wonseok Jeon, Junyoung Park, Mingu Lee, and Christopher Lott. 2024. On Speculative Decoding for Multimodal Large Language Models. *arXiv preprint arXiv:2404.08856*.

Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Rohan Gemini Team Google: Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1233–1239.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. 2024. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to Describe Differences Between Pairs of Similar Images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yizhang Jin, Jian Li, Yexin Liu, Tianjun Gu, Kai Wu, Zhengkai Jiang, Muyang He, Bo Zhao, Xin Tan, Zhenye Gan, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. 2024. Efficient Multimodal Large Language Models: A Survey. *Preprint*, arXiv:2405.10739.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. LLaVA-OneVision: Easy Visual Task Transfer. *arXiv preprint arXiv:2408.03326*.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024b. LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models. *arXiv preprint arXiv:2407.07895*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating Object Hallucination in Large Vision-Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6329–6338.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024c. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Jian Liang, Ran He, and Tieniu Tan. 2025. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2024. Boosting Multimodal Large Language Models with Visual Tokens Withdrawal for Rapid Inference. *arXiv preprint arXiv:2405.05803*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*.

Shuo Liu, Kaining Ying, Hao Zhang, Yue Yang, Yuqi Lin, Tianle Zhang, Chuanhao Li, Yu Qiao, Ping Luo, Wenqi Shao, and Kaipeng Zhang. 2024b. Convbench: A multi-turn conversation evaluation benchmark with hierarchical capability for large vision-language models. *Preprint*, arXiv:2403.20194.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2024c. Ocrbench: On the hidden mystery of ocr in large multimodal models. *Preprint*, arXiv:2305.07895.

Zuyan Liu, Benlin Liu, Jiahui Wang, Yuhao Dong, Guangyi Chen, Yongming Rao, Ranjay Krishna, and Jiwen Lu. 2024d. Efficient Inference of Vision Instruction-Following Models with Elastic Cache. *arXiv preprint arXiv:2407.18121*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279.

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.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. 2024. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023. Specinfer: Accelerating generative llm serving with speculative inference and token tree verification. *arXiv preprint arXiv:2305.09781*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

QwenTeam. Qwen Speed Benchmark. `https://qwen.readthedocs.io/en/latest/benchmark/speed_benchmark.html`. Accessed: 2024-11-10.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

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*.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: a dataset for image captioning with reading comprehension. *Preprint*, arXiv:2003.12462.

Shezheng Song, Xiaopeng Li, Shasha Li, Shan Zhao, Jie Yu, Jun Ma, Xiaoguang Mao, and Weimin Zhang. 2023. How to bridge the gap between modalities: A comprehensive survey on multimodal large language model. *Preprint*, arXiv:2311.07594.

Hanshi Sun, Zhuoming Chen, Xinyu Yang, Yuandong Tian, and Beidi Chen. 2024a. Triforce: Lossless acceleration of long sequence generation with hierarchical speculative decoding. *arXiv preprint arXiv:2404.11912*.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. 2024b. SpecTr: Fast Speculative Decoding via Optimal Transport. *Preprint*, arXiv:2310.15141.

Hao Tan, Franck Dernoncourt, Zhe Lin, Trung Bui, and Mohit Bansal. 2019. Expressing Visual Relationships via Language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1873–1883.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. 2024. Accelerating auto-regressive text-to-image generation with training-free speculative jacobi decoding. *arXiv preprint arXiv:2410.01699*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. LOOK-M: Look-Once Optimization in KV Cache for Efficient Multimodal Long-Context Inference. *arXiv preprint arXiv:2406.18139*.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S. Yu. 2023. Multimodal Large Language Models: A Survey. *Preprint*, arXiv:2311.13165.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *Preprint*, arXiv:2401.07851.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829.

Sen Yang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2024. Multi-Candidate Speculative Decoding. *arXiv preprint arXiv:2401.06706*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A Survey on Multimodal Large Language Models. *Preprint*, arXiv:2306.13549.

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*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid Loss for Language Image Pre-Training. *Preprint*, arXiv:2303.15343.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. 2024a. MM-LLMs: Recent Advances in MultiModal Large Language Models. *Preprint*, arXiv:2401.13601.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. 2024b. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2024. DistillSpec: Improving Speculative Decoding via Knowledge Distillation. In *The Twelfth International Conference on Learning Representations*.

Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. CRC press.

A TABED with Different Draft Models

In this section, we evaluate the performance of our TABED for two different choices of the draft model: LLaVA-1.5 160M (fine-tuned with the same recipe as LLaVA-1.5 68M), and LLaVA-OV (the same architecture as LLaVA-1.5 68M but fine-tuned with the multi-image-aware OneVision recipe). The full results are presented in Table 3.

B Details for Caption Drafting

In this section, we describe various types of lightweight image captioning models that can be used for caption drafting (Appendix B.1). We then demonstrate that captioning model inference completes earlier than the target model's prefilling by analyzing the captioning model's latency (Appendix B.2).

B.1 Captioning Models

BLIP (Li et al., 2022) A vision-language model trained on bootstrapped synthetic captions. It uses a visual transformer and the text encoder of BERT (Devlin et al., 2019) to separately encode image and text.

`https://huggingface.co/Salesforce/blip-image-captioning-base`

BLIP-2 (Li et al., 2023a) A vision-language model using a frozen off-the-shelf image encoder and LLM. A querying transformer trained using boostrapped data is included for cross-modal alignment.

`https://huggingface.co/Salesforce/blip2-opt-2.7b`

Florence-2 (Xiao et al., 2024) A vision-language model that is instruction-trained for a variety of tasks. Its architecture consists of a single sequence-to-sequence transformer and a vision encoder.

`https://huggingface.co/microsoft/Florence-2-large-ft`

B.2 Latency Analysis

It is important to ensure that the captioning model runs fast enough so that it does not delay drafting. In this line, we measure in Table 4 the time taken by the two captioning models, BLIP and Florence-2, to generate captions. The results demonstrate captioning completes earlier than target model's prefilling.

Draft Model			Benchmark Datasets							OOD Datasets	
Type	Size	Method	LLaVA-W	DocVQA	POPE	MMVet	IEdit	MB	Spot	PSV	VIST
LLaVA-OV	68M	M	1.97	2.02	2.20	2.05	2.07	2.15	2.12	1.77	1.74
		T	2.00	2.04	2.07	2.05	2.19	2.25	2.11	1.90	1.83
		MT	2.02	2.08	2.26	2.08	2.24	2.32	2.22	1.96	1.91
LLaVA-1.5	160M	M	2.61	2.47	2.89	2.52	2.43	2.23	2.66	1.29	1.27
		T	2.47	2.31	2.50	2.42	2.56	2.73	2.63	2.28	2.31
		MT	2.60	2.49	2.80	2.54	2.65	2.74	2.78	2.18	2.15

Table 3: Evaluation of a larger LLaVA-1.5 draft model (160M), obtained through standard visual instruction tuning, and a same-sized LLaVA-OV draft model (68M), obtained through multi-image-aware fine-tuning.

		Latency (s)		
Model	Type	$n = 1$	$n = 2$	$n = 5$
Target LVLM (prefilling)	LLaVA-1.5 7B	0.112	0.207	0.540
Image Captioning	BLIP	**0.054**	**0.055**	**0.074**
	Florence-2	0.105	0.149	0.292

Table 4: Latency analysis of image captioning models. BLIP and Florence-2 captioning latencies are lower than the target LVLM's prefilling latency. Parallel processing can therefore hide captioning latency without affecting time to first token.

	Datasets	
Model	Spot-the-Diff	MagicBrush
GPT-4o	19.1	10.5
GPT-4o-mini	12.0	9.1
LLaVA-1.5 7B	13.9	9.6
LLaVA-1.5 7B (text-only)	11.1	8.4

Table 5: Evaluation results for the target model on benchmark datasets. Our target model, LLaVA-1.5 7B, demonstrates performance comparable to GPT-4o-mini.The injection of multimodal information in the input scenarios of the benchmark datasets benefits the target model, LLaVA-1.5 7B.

C Evaluation of Target Model

In this section, we present both quantitative results (Appendix C.1) and qualitative samples (Appendix C.2) produced by LLaVA-1.5 7B, which serves as the target model in our experimental setting.

C.1 Quantitaive Evaluation

We assess the output quality of LLaVA-1.5 7B using benchmark datasets, as shown in Table 5. The results demonstrate that the quality of LLaVA-1.5 7B is comparable to GPT-4o-mini.The injection of multimodal information benefits the target model, LLaVA-1.5 7B, from the perspective of output quality.

C.2 Generated Samples

Spot The Difference [Fig. 6e]

Q: What has changed from the first image to the second?

A: The image shows a city street with a grassy area in the middle of the road. In the first image, there are several cars parked along the street, and a few people are walking around. The street appears to be relatively empty, with no traffic or significant activity. In the second image, the street is bustling with activity. There are numerous cars parked along the street, and several people are walking around. The scene is more lively and dynamic, with a higher level of activity compared to the first image.

MagicBrush [Fig. 6f]

Q: Please provide the image edit instruction that can transfrom the source image to the target image.

A: To transform the source image into the target image, follow these steps: 1. Remove the red-haired girl from the scene, as she is not part of the target image. 2. Replace the girl with a back pack, which should be placed on the stairs. 3. Add a bus in the background, making sure it is yellow and occupies the entire width of the image. 4. Place a person in the background, standing near the bus, to give a sense of scale and context. 5. Adjust the lighting and shadows to create a more realistic scene.

PororoSV [Fig. 6g]

Q: Taking the stories from the initial images into account, can you conclude the story using the elements of the final image?

A: The image features a group of cartoon characters, including a penguin, a fox, and a pink character, standing in a snowy environment. The characters are positioned in various locations, with some standing closer to the foreground and others further back. The scene appears to be a part of a video game or animated series, as the characters are interacting with each other and their surroundings.

VIST [Fig. 6h]

Q: With the narratives paired with the initial images, how would you conclude the story using the last picture?

A: The image captures a thrilling moment during a volley ball game, with a woman leaping into the air to hit the ball. She is wearing a purple shirt and is in the process of jumping to make contact with the sports ball. The scene is filled with excitement and anticipation as the players on the court watch her performance. The volley ball game is taking place on a court, and the players are actively engaged in the match.

D Empirical Validation of Remarks 1-3

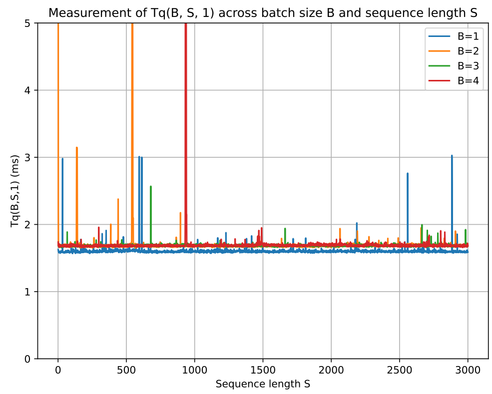


Figure 3: Empirical demonstration of Remarks 2 and 3.

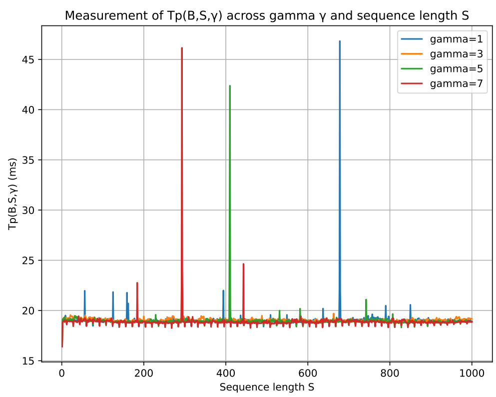


Figure 4: Empirical demonstration of Remark 1.

In this section, we validate the Remarks from Sec. 3.1 with empirical timing measurements. Focusing on the case where the remarks apply to speculative decoding settings, we use LLaVA-1.5 7B and LLaVA-1.5 68M to measure $T_p(B, S, \gamma)$ for Remark 1 and $T_q(B, S, 1)$ for Remarks 2 and 3, respectively. All experiments are conducted on an A100 80GB GPU using the fp16 data type for the models. Under moderate S (*e.g.*, $S \leq$ 3k) and sufficiently small B (*e.g.*, $B \leq 4$) and γ (*e.g.*, $\gamma \leq 10$), the decoding phase displays the following observations:

Remark 1. *For given B and S, regardless of γ, $T(B, S, \gamma)$ remains approximately constant* (e.g., *$|\Delta T/T| < 0.05$).*

Remark 2. *For a given B, regardless of S, $T(B, S, 1)$ remains approximately constant* (e.g., *$|\Delta T/T| < 0.05$).*

Remark 3. *For a given S, regardless of B, $T(B, S, 1)$ remains approximately constant* (e.g., *$|\Delta T/T| < 0.05$).*

Fig. 3 shows $T_q(B, S, 1)$ in milliseconds for sequence lengths up to 3k for each batch size $B \in \{1, 2, 3, 4\}$. For moderate sequence lengths $S \leq$ 3k, T_q varies by no more than 5% for each B, which supports Remark 2. Similarly, when comparing different Bs with a fixed S, T_q varies by no more than 5%, which supports Remark 3.

Fig. 4 shows $T_p(B, S, \gamma)$ in milliseconds for each $\gamma \in \{1, 3, 5, 7\}$. We test the case of B=1, which aligns with our experimental settings where the target model always performs inference on a single batch. Over the values of γ considered, T_p varies by no more than 5%.

E Training and Evaluation of Draft Models

In this section, we present a more detailed overview of our custom training procedure for the draft models (Appendix E.1). We then evaluate our primary draft model, LLaVA-1.5 68M, on multimodal tasks to ensure it has the capability to properly perceive multimodality, and we provide some qualitative samples from the draft model (Appendix E.2).

E.1 Details of Training

LLaVA-1.5 (Liu et al., 2024a) The process for developing draft models with LLaVA-1.5 (68M, 160M) training recipe was divided into two stages: pre-training and instruction fine-tuning (IFT). Pre-training focuses on training the projector while the parameters of the LLM and vision encoder are frozen. During the IFT stage, visual instruction tuning is used to teach the LLM to follow multimodal instructions. The vision encoder remains frozen throughout both stages. The hyperparameters used for each stage are described in Table 6. We trained the draft model using datasets curated by the original author of LLaVA-1.5. For more training details, see https://github.com/haotian-liu/LLaVA/tree/main.

Hyperparameter	Value
Training Epochs	1
Batch Size	256
Learning Rate (LR)	1e-3
LR Schedule Type	Cosine
Warm-up Ratio	0.03
Weight Decay	0.0

(a) Pretraining stage

Hyperparameter	Value
Training Epochs	1
Batch Size	128
Learning Rate (LR)	2e-5
LR Schedule Type	Cosine
Warm-up Ratio	0.03
Weight Decay	0.0

(b) Instruction fine-tuning stage

Table 6: Details of hyperparameters used in LLaVA-1.5 training

LLaVA-OneVision (Li et al., 2024a) The development of draft models using the LLaVA-OneVision (LLaVA-OV) training recipe was divided into three stages: language-image alignment, high-quality knowledge learning, and visual instruction tuning. In the language-image alignment stage, visual features are aligned with the word embedding space of LLMs. High-quality knowledge learning balances computational efficiency with the integration of new knowledge into LVLMs. Visual instruction tuning consists of two phases: (i) Single-Image Training, where the model learns to perform visual tasks using instructions from single images, and (ii) OneVision Training, where the model learns to execute multi-image visual tasks using a blend of video, single-image, and multi-image data. During the language-image alignment stage, only the projector for aligning visual features is updated, whereas all components including LLM are updated in the following three stages. We trained the draft model using datasets curated by the original author of LLaVA-OV (Li et al., 2024a). The hyperparameters used for each stage are described in Table 7, and the learning rate for the vision encoder is one-fifth of that for the LLM across all stages. For more details, visit https://github.com/LLaVA-VL/LLaVA-NeXT.

E.2 Evaluation Results

Table 8 presents the evaluation results of our primary draft model, LLaVA-1.5 68M, on OCRBench (Liu et al., 2024c) and TextCaps (Sidorov et al., 2020) datasets. We assess the output quality of the draft model with and without image inputs and compare the results with those of the target model, LLaVA-1.5 7B. In terms of output quality, the draft model with image inputs consistently outperforms the one without, illustrating that the injection of multimodal information benefits the custom-trained draft model.

Fig. 5 presents qualitative samples from the OCRBench dataset. Both LLaVA-1.5 7B and 68M models provided accurate responses, whereas the text-only LLaVA-1.5 68M model failed to answer correctly due to its lack of image-processing capabilities.

Hyperparameter	Value
Training Epochs	1
Batch Size	512
Learning Rate (LR)	1e-3
LR Schedule Type	Cosine
Warm-up Ratio	0.03
Weight Decay	0.0

(a) Image alignment stage

Hyperparameter	Value
Training Epochs	1
Batch Size	512
Learning Rate (LR)	1e-5
LR Schedule Type	Cosine
Warm-up Ratio	0.03
Weight Decay	0.0

(b) High-quality knowledge learning stage

Hyperparameter	Value
Training Epochs	1
Batch Size	512
Learning Rate (LR)	1e-5
LR Schedule Type	Cosine
Warm-up Ratio	0.03
Weight Decay	0.0

(c) Visual instruction tuning stage: Single-image training

Hyperparameter	Value
Training Epochs	1
Batch Size	512
Learning Rate (LR)	1e-5
LR Schedule Type	Cosine
Warm-up Ratio	0.03
Weight Decay	0.0

(d) Visual instruction tuning stage: OneVision training

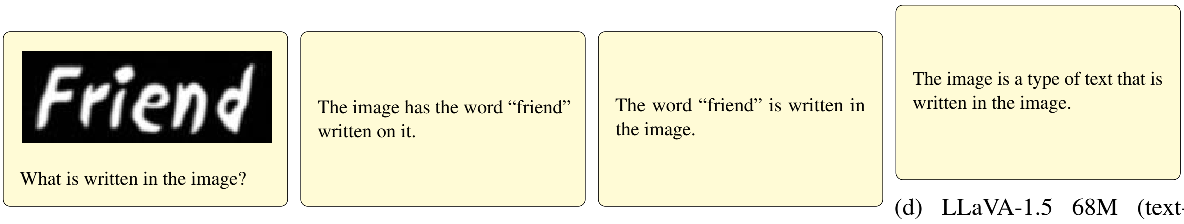
Table 7: Details of hyperparameters used in LLaVA-OV training

	OCRBench	TextCaps	
Model	Accuracy	METEOR	ROUGE
LLaVA-1.5 7B	0.207	0.249	0.480
LLaVA-1.5 68M	0.048	0.133	0.254
LLaVA-1.5 68M (text-only)	0.014	0.064	0.132

Table 8: Evaluation results for the off-the-shelf target model and the custom-trained draft model on MLLM tasks. Since the draft model is trained to perceive multi-modality, the injection of multimodal information benefits the custom-trained draft model.

F Prompts for Each Dataset and Drafting

In this section, we describe the formats of prompts used for inference on each dataset, including system prompts and how to organize prompts with text and image inputs (Appendix F.1). We then provide details on replacing image tokens in text-only and caption drafting (Appendix F.2).



(a) Instruction (b) LLaVA-1.5 7B (c) LLaVA-1.5 68M (d) LLaVA-1.5 68M (text-only)

Figure 5: Qualitative evaluation samples from the OCRBench dataset by LLaVA-1.5 7B and 68M. Both the target (b) and the draft (c) models recognize the text “friend” written on the image by multimodal reasoning whereas the text-only model (d) fails, as expected.

F.1 Prompt Formats for Each Dataset

We use the following prompt formats for respective tasks. Based on the template for chat (*USER:* and *ASSISTANT:*), each system prompt is prepended with the start token *<s>*. The *<image>* token is used to represent image data within a prompt. *[QUESTION]* and *[CAPTION]* are placeholders denoting information unique to each sample of a dataset.

LLaVA-Bench (In-the-Wild) *<s> USER: <image> For the following question, provide a detailed explanation of your reasoning leading to the answer. [QUESTION] ASSISTANT:*

DocVQA *<s> USER: <image> For the following question, provide a detailed explanation of your reasoning leading to the answer. [QUESTION] ASSISTANT:*

POPE *<s> USER: <image> For the following question, provide a detailed explanation of your reasoning leading to the answer. [QUESTION] ASSISTANT:*

MMVet *<s> USER: <image> For the following question, provide a detailed explanation of your reasoning leading to the answer. [QUESTION] ASSISTANT:*

IEdit *<s> USER: Please provide instructions for editing the source image to match the target image. Source Image: <image> Target Image: <image> Instruction: ASSISTANT:*

MagicBrush *<s> USER: Please provide instructions for editing the source image to match the target image. Source Image: <image> Target Image: <image> Instruction: ASSISTANT:*

Spot The Difference *<s> USER: Explain the disparities between the first and second image. <image> <image> Difference: ASSISTANT:*

PororoSV *<s> USER: Given the progression of the story with the first few images, can you write a fitting end considering the last image? <image> Caption #1: [CAPTION] <image> Caption #2: [CAPTION]. <image> Caption #3: [CAPTION] <image> Caption #4: [CAPTION] <image> Caption #5: ASSISTANT:*

VIST *<s> USER: With the narratives paired with the initial images, how would you conclude the story using the last picture? <image> Caption #1: [CAPTION] <image> Caption #2: [CAPTION]. <image> Caption #3: [CAPTION] <image> Caption #4: [CAPTION] <image> Caption #5: ASSISTANT:*

F.2 Replacing Image tokens in Draftings

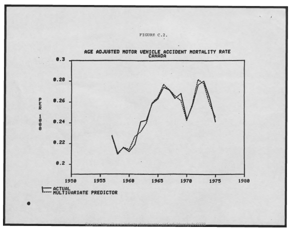
For text-only drafting, the *<image>* token is replaced by the escape character “\n”. We experimented with several replacement methods: (1) tokenizing the string “<image>” into three tokens, and (2) retaining the special token <image> without replacing it with an image embedding. Method (2) resulted in very poor block efficiency, but method (1) showed comparable block efficiency. Our replacement approach is simple because it ensures that the prompt length remains consistent before and after replacement.

For caption drafting, the *<image>* token is replaced by a generated caption with a prefix. Specifically, after the lightweight captioning model generates a caption based on the image inputs in the sample, we prepend the string “image: ” to the caption and replace the *<image>* token.

G Details of Each Dataset

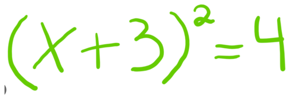
In this section, we describe each of the curated datasets in benchmark (Appendix G.1) and OOD (Appendix G.2) datasets and provide links to them for convenience and reproducibility.

(a) LLaVA-Bench (In-the-Wild)



(b) DocVQA

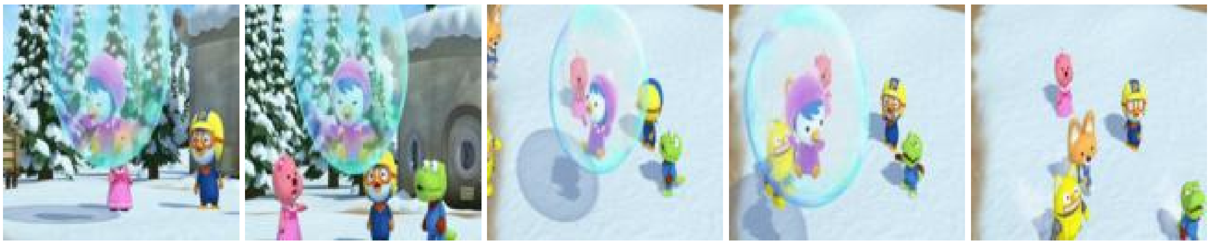
(c) POPE



(d) MMVET

(e) Spot the Difference

(f) MagicBrush



(g) PororoSV

(h) VIST

Figure 6: Qualitative image samples of benchmark and OOD datasets. The corresponding questions and answers are presented in Appendix C.

G.1 Benchmark Datasets

LLaVA-Bench (In-the-Wild) (Liu et al., 2023) A dataset for comparing the performance of vision-language models against state-of-the-art proprietary models. Each prompt is provided with an image, a caption, and a reference answer from text-only GPT-4. Prompt styles include question answering, image description, and complex reasoning. In total, the dataset contains 60 unique prompts and 24 unique images.

https://huggingface.co/datasets/lmms-lab/llava-bench-in-the-wild

DocVQA (Mathew et al., 2021) DocVQA is a dataset designed for visual question answering on document images, comprising 50,000 questions over 12,000+ diverse document images.

https://huggingface.co/datasets/lmms-lab/LMMs-Eval-Lite

POPE (Li et al., 2023b) A multimodal question answering dataset that asks binary (yes or no) questions about whether certain objects are present in an image. The subset used for evaluation in our work contains 100 pairs of images and questions.

https://huggingface.co/datasets/lmms-lab/LMMs-Eval-Lite

MMVet (Yu et al., 2023) MM-Vet is a benchmark designed to evaluate large multimodal models on complex tasks, focusing on the integration of six core vision-language capabilities: recognition, OCR, knowledge, language generation, spatial awareness, and math.

https://huggingface.co/datasets/lmms-lab/MMVet

Spot the Difference (Jhamtani and Berg-Kirkpatrick, 2018) A dataset of crowd-sourced descriptions of differences between a pair of images. The subset used for evaluation in our work contains 100 annotated image pairs collected using individual frames of security-footage data.

https://huggingface.co/datasets/lmms-lab/LLaVA-NeXT-Interleave-Bench

IEdit (Tan et al., 2019) A dataset to train models to describe the relationship between images via editing instructions. The subset used for evaluation in our work contains 100 image pairs of a source image and a target image, accompanied by instructions on how to transform the source image into the target.

https://huggingface.co/datasets/lmms-lab/LLaVA-NeXT-Interleave-Bench

MagicBrush (Zhang et al., 2024b) A dataset for text-guided image editing containing manually annotated editing instructions to transform one real image into another. The subset used for evaluation in our work contains 100 triplets of a source image, a target image, and editing instructions.

https://huggingface.co/datasets/lmms-lab/LLaVA-NeXT-Interleave-Bench

G.2 OOD Datasets

Pororo-SV (Li et al., 2019) A dataset of stories each created by pairing 5 consecutive frames from the animated series *Pororo* with a text description. The subset used for evaluation in our work contains 100 stories.

https://huggingface.co/datasets/lmms-lab/LLaVA-NeXT-Interleave-Bench

VIST (Huang et al., 2016) A dataset of sequential images paired with three types of descriptions ranging from isolated factual descriptions to causal, narrative interpretations. The subset used for evaluation in our work contains 100 sequences of 3 images.

https://huggingface.co/datasets/lmms-lab/LLaVA-NeXT-Interleave-Bench

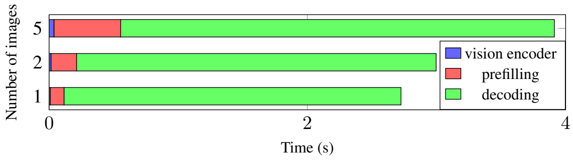


Figure 7: Inference time analysis for the LLaVA-1.5 7B model. Although the time for vision encoder and prefilling increases with the number of images, the decoding stage still dominates.

G.3 Time Analysis of LVLM Inference Stages

To analyze how the number of input images affect the LVLM inference time, we select ChartQA (Masry et al., 2022), Spot the Difference (Jhamtani and Berg-Kirkpatrick, 2018), and PororoSV (Li et al., 2019) datasets representing 1, 2, and 5 images with corresponding visual context lengths of 0.6k, 1.2k, and 3k, respectively. We visualize the generation time by component in Fig. 7 with 100 generated tokens for analysis, with actual average decoding lengths of 92, 117, and 88, respectively. The execution time of the *vision encoder* and *prefilling* stages increases in proportion with the number of input images, as each image is converted into several hundred context tokens. In contrast, the *decoding* stage shows little difference in execution time across varying numbers of input images, while dominating the total generation time. Hence, although reducing the number of visual tokens (Shang et al., 2024; Chen et al., 2024b; Lin et al., 2024) would significantly improve the efficiency of *vision encoder* and *prefilling* stages, it would have only marginal impact on the dominant *decoding* stage.