Mitigating Sequential Dependencies: A Survey of Algorithms and Systems for Generation-Refinement Frameworks in Autoregressive Models

Anonymous EMNLP submission

Abstract

Sequential dependencies present a fundamental bottleneck in deploying large-scale autoregressive models, particularly for real-time applications. While traditional optimization approaches like pruning and quantization often compromise model quality, recent advances in generation-refinement frameworks demonstrate that this trade-off can be significantly mitigated.

This survey presents a comprehensive taxonomy of generation-refinement frameworks, analyzing methods across autoregressive sequence tasks. We categorize methods based on their generation strategies (from simple n-gram prediction to sophisticated draft models) and refinement mechanisms (including single-pass verification and iterative approaches). Through systematic analysis of both algorithmic innovations and system-level implementations, we examine deployment strategies across computing environments and explore applications spanning text, images, and speech generation. This systematic examination of both theoretical frameworks and practical implementations provides a foundation for future research in efficient autoregressive decoding. In the appendixA, we additionally provide experimental comparisons of various baseline methods.

1 Introduction

Large Models (LMs) have demonstrated remarkable capabilities across diverse domains, from text generation (Brown et al., 2020; Zhuang et al., 2023; Touvron et al., 2023) and translation (Zhu et al., 2023; Hadi et al., 2023; Huang et al., 2023) to image synthesis (Ho et al., 2020; Yang et al., 2023a; Tian et al., 2024) and video generation (Ding et al., 2023; Wu et al., 2023; ope, 2024). However, these models face a critical challenge: their inherently sequential nature creates significant latency bottlenecks, particularly for real-time applications. While traditional optimization approaches like quantization and pruning often compromise model quality for speed, recent research has focused on maintaining output quality while breaking sequential dependencies through novel algorithmic and system-level innovations.

Generation-refinement frameworks have emerged as a promising family of solutions that directly address these sequential bottlenecks. These approaches encompass a range of methods, from speculative decoding with draft models to iterative refinement techniques inspired by numerical optimization. The common thread among these approaches is their division of the generation process into two phases: an initial generation step that produces draft tokens in parallel, followed by a refinement step that ensures output quality.

The implementation of these frameworks presents unique system-level challenges across different deployment scenarios. Edge devices require careful optimization of memory usage and computation patterns (Svirschevski et al., 2024; Xu et al., 2024a), while distributed systems must manage complex communication patterns and load balancing. These system-level considerations have driven innovations in areas like kernel design, hardware acceleration, and batch processing optimization, significantly influencing both algorithmic choices and practical performance.

This survey synthesizes research across these approaches, examining both algorithmic innovations and their system implementations. We present a systematic taxonomy of generation-refinement methods, analyze deployment strategies across computing environments, and explore applications spanning text, images (Wang et al., 2024d; Jang et al., 2024), and speech (Li et al., 2024a; Raj et al., 2024). Our contributions include comprehensive analysis of system-level implementations and optimizations, detailed examination of applications across modalities, and identification of key research

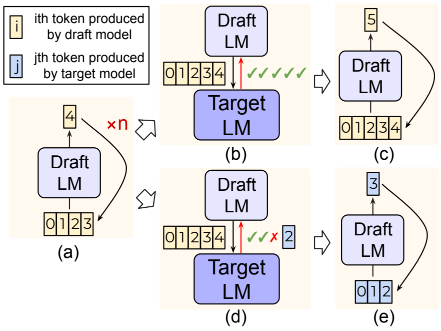

i ith token produced by draft model
j jth token produced by target model
Draft LM
Target LM
0 1 2 3 4
✓✓✓✓✓
×n
(a)
(b)
(c)
(d)
(e)
✓✓✗ 2
0 1 2 3

Figure 1: Illustration of speculative decoding workflow.

challenges in efficient neural sequence generation.

2 The Sequential Bottleneck in Large Model Inference

Traditional approaches to accelerating LM inference have focused on reducing computational costs through model compression, knowledge distillation, and architectural optimizations. However, these methods primarily address individual computation costs rather than the fundamental sequential dependency that requires each token to wait for all previous tokens.

Speculative decoding (SD) (Stern et al., 2018) has emerged as a promising solution that directly targets this sequential bottleneck. As illustrated in Figure 1, this approach introduces a two-phase process where a smaller, faster *draft model* first predicts multiple tokens in parallel, followed by verification using the target model. The draft model enables parallel token generation, breaking away from traditional token-by-token generation, while the target model's verification step maintains output quality through accept/reject decisions.

This strategy has proven particularly valuable for real-time applications like interactive dialogue systems, where response latency directly impacts user experience. The verification mechanism provides a crucial balance between generation speed and output quality, accepting correct predictions to maintain throughput while falling back to sequential generation when necessary to preserve accuracy.

While SD represents one successful approach to breaking sequential dependencies in autoregressive (AR) models, it belongs to a broader family of *generation-refinement* methods. The following sections present a systematic taxonomy of these approaches, examining how different techniques balance the trade-offs between generation parallelism and output quality.

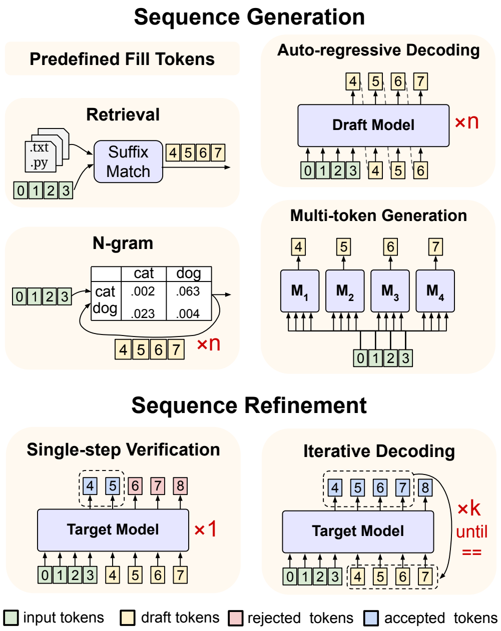


Figure 2: A taxonomy of generation-refinement frameworks, showing two phases: (1) Generation of draft tokens through various methods and (2) Refinement through verification strategies.

3 A Taxonomy for Generation and Refinement Frameworks

To systematically analyze approaches for breaking sequential dependencies in large models, we propose a unified taxonomy that categorizes methods based on their generation and refinement strategies. As shown in Figure 2, our taxonomy decomposes these frameworks into two fundamental phases: *Sequence Generation* and *Sequence Refinement*. This decomposition not only encompasses traditional SD approaches but also captures a broader range of emerging methods that trade off between generation parallelism and output quality.

The sequence generation phase focuses on different strategies for producing draft tokens more efficiently than conventional auto-regressive decoding using a single larger model. These strategies range from simple approaches like random token sampling (used in conjunction with iterative decoding) to more sophisticated methods like retrieval-based generation and draft model prediction. Each generation method offers trade-offs in terms of computational cost and prediction quality. The sequence refinement phase then determines how these candidates are processed - either accepting them directly (with possible poorer quality), verifying a subset of tokens in a single pass, or refining the draft tokens through multiple iterations until convergence.

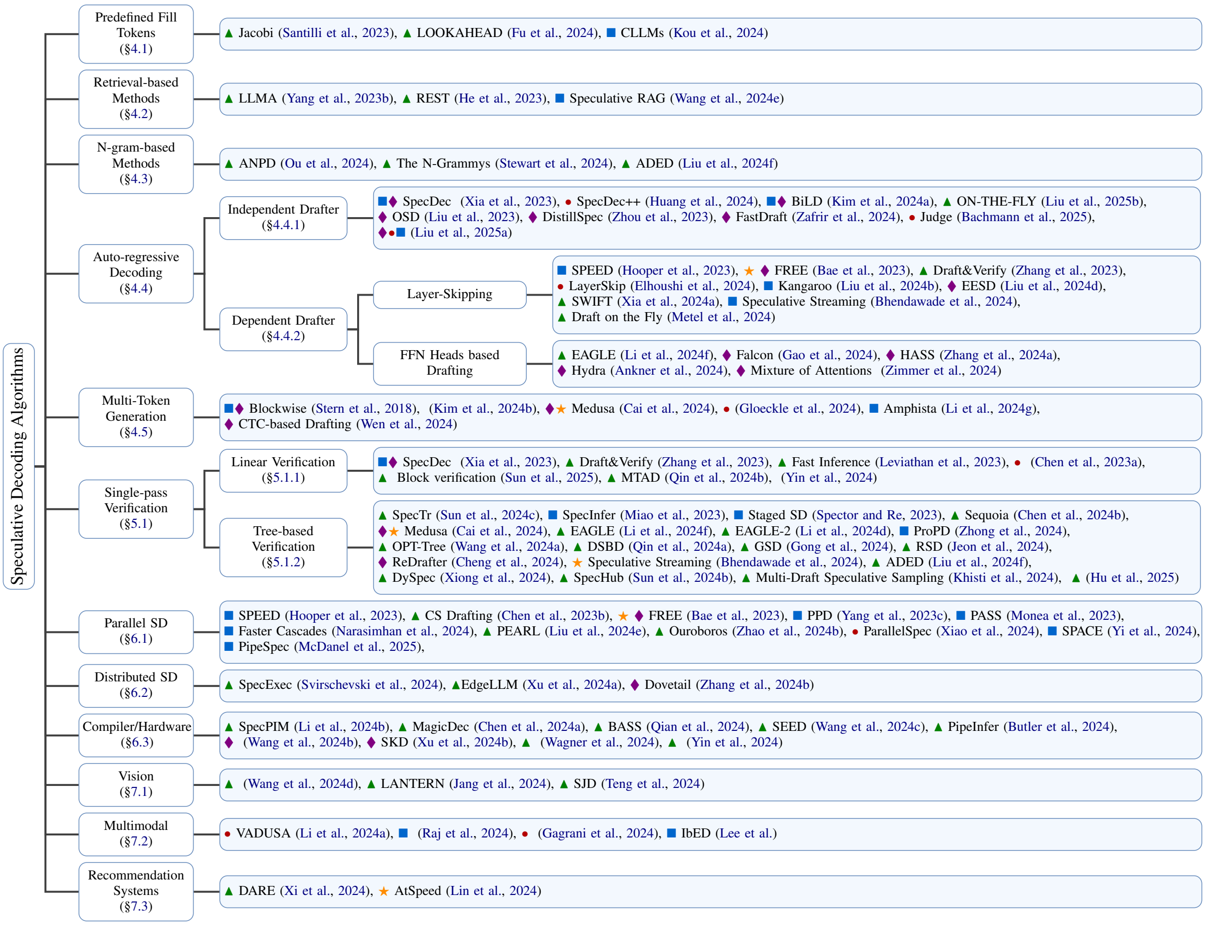

Speculative Decoding Algorithms
Predefined Fill Tokens (§4.1)
▲ Jacobi (Santilli et al., 2023), ▲ LOOKAHEAD (Fu et al., 2024), ■ CLLMs (Kou et al., 2024)
Retrieval-based Methods (§4.2)
▲ LLMA (Yang et al., 2023b), ▲ REST (He et al., 2023), ■ Speculative RAG (Wang et al., 2024e)
N-gram-based Methods (§4.3)
▲ ANPD (Ou et al., 2024), ▲ The N-Grammys (Stewart et al., 2024), ▲ ADED (Liu et al., 2024f)
Auto-regressive Decoding (§4.4)
Independent Drafter (§4.4.1)
■♦ SpecDec (Xia et al., 2023), ● SpecDec++ (Huang et al., 2024), ■♦ BiLD (Kim et al., 2024a), ▲ ON-THE-FLY (Liu et al., 2025b), ♦ OSD (Liu et al., 2023), ♦ DistillSpec (Zhou et al., 2023), ♦ FastDraft (Zafrir et al., 2024), ● Judge (Bachmann et al., 2025), ♦●■ (Liu et al., 2025a)
Dependent Drafter (§4.4.2)
Layer-Skipping
■ SPEED (Hooper et al., 2023), ★ ♦ FREE (Bae et al., 2023), ▲ Draft&Verify (Zhang et al., 2023), ● LayerSkip (Elhoushi et al., 2024), ■ Kangaroo (Liu et al., 2024b), ♦ EESD (Liu et al., 2024d), ▲ SWIFT (Xia et al., 2024a), ■ Speculative Streaming (Bhendawade et al., 2024), ▲ Draft on the Fly (Metel et al., 2024)
FFN Heads based Drafting
▲ EAGLE (Li et al., 2024f), ♦ Falcon (Gao et al., 2024), ♦ HASS (Zhang et al., 2024a), ♦ Hydra (Ankner et al., 2024), ♦ Mixture of Attentions (Zimmer et al., 2024)
Multi-Token Generation (§4.5)
■♦ Blockwise (Stern et al., 2018), (Kim et al., 2024b), ♦★ Medusa (Cai et al., 2024), ● (Gloeckle et al., 2024), ■ Amphista (Li et al., 2024g), ♦ CTC-based Drafting (Wen et al., 2024)
Single-pass Verification (§5.1)
Linear Verification (§5.1.1)
■♦ SpecDec (Xia et al., 2023), ▲ Draft&Verify (Zhang et al., 2023), ▲ Fast Inference (Leviathan et al., 2023), ● (Chen et al., 2023a), ▲ Block verification (Sun et al., 2025), ▲ MTAD (Qin et al., 2024b), (Yin et al., 2024)
Tree-based Verification (§5.1.2)
▲ SpecTr (Sun et al., 2024c), ■ SpecInfer (Miao et al., 2023), ■ Staged SD (Spector and Re, 2023), ▲ Sequoia (Chen et al., 2024b), ♦★ Medusa (Cai et al., 2024), ▲ EAGLE (Li et al., 2024f), ▲ EAGLE-2 (Li et al., 2024d), ■ ProPD (Zhong et al., 2024), ▲ OPT-Tree (Wang et al., 2024a), ▲ DSBD (Qin et al., 2024a), ▲ GSD (Gong et al., 2024), ▲ RSD (Jeon et al., 2024), ♦ ReDrafter (Cheng et al., 2024), ★ Speculative Streaming (Bhendawade et al., 2024), ▲ ADED (Liu et al., 2024f), ▲ DySpec (Xiong et al., 2024), ▲ SpecHub (Sun et al., 2024b), ▲ Multi-Draft Speculative Sampling (Khisti et al., 2024), ▲ (Hu et al., 2025)
Parallel SD (§6.1)
■ SPEED (Hooper et al., 2023), ▲ CS Drafting (Chen et al., 2023b), ★ ♦ FREE (Bae et al., 2023), ■ PPD (Yang et al., 2023c), ■ PASS (Monea et al., 2023), ■ Faster Cascades (Narasimhan et al., 2024), ▲ PEARL (Liu et al., 2024e), ▲ Ouroboros (Zhao et al., 2024b), ● ParallelSpec (Xiao et al., 2024), ■ SPACE (Yi et al., 2024), ■ PipeSpec (McDanel et al., 2025),
Distributed SD (§6.2)
▲ SpecExec (Svirschevski et al., 2024), ▲EdgeLLM (Xu et al., 2024a), ♦ Dovetail (Zhang et al., 2024b)
Compiler/Hardware (§6.3)
▲ SpecPIM (Li et al., 2024b), ▲ MagicDec (Chen et al., 2024a), ▲ BASS (Qian et al., 2024), ▲ SEED (Wang et al., 2024c), ▲ PipeInfer (Butler et al., 2024), ♦ (Wang et al., 2024b), ♦ SKD (Xu et al., 2024b), ▲ (Wagner et al., 2024), ▲ (Yin et al., 2024)
Vision (§7.1)
▲ (Wang et al., 2024d), ▲ LANTERN (Jang et al., 2024), ▲ SJD (Teng et al., 2024)
Multimodal (§7.2)
● VADUSA (Li et al., 2024a), ■ (Raj et al., 2024), ● (Gagrani et al., 2024), ■ IbED (Lee et al.)
Recommendation Systems (§7.3)
▲ DARE (Xi et al., 2024), ★ AtSpeed (Lin et al., 2024)

Figure 3: Taxonomy of Speculative Decoding Algorithms. Symbols indicate implementation approach: ▲ Direct application (no training required), ● Full model training from scratch, ■ Model fine-tuning, ★ Parameter-efficient fine-tuning (PEFT), ♦ Knowledge distillation from target model.

4 Sequence Generation Methods

4.1 Predefined Fill Tokens

The simplest approach uses random initialization or predefined tokens (e.g., PAD). While computationally free, these methods provide poor initialization points, requiring multiple refinement iterations as discussed in Section 5.2.

4.2 Retrieval-based Methods

LLMA (Yang et al., 2023b) first proposed exploiting overlaps between LLM outputs and reference documents to accelerate inference through parallel token verification while maintaining identical generation results. In retrieval-based approaches, REST (He et al., 2023) replaces smaller language models with exact suffix matching from a datastore to generate draft tokens. It builds a Trie (prefix tree) from retrieved continuations, where node weights reflect token sequence frequencies. Speculative RAG (Wang et al., 2024e) use a fine-tuned specialist LM to generate complete answer drafts with supporting rationales. It clusters retrieved documents by similarity, generates diverse drafts from different document subsets, and employs self-consistency and self-reflection scores for draft evaluation instead of token-level verification.

4.3 N-gram-based Methods

Several approaches leverage n-gram patterns for efficient token generation. ANPD (Ou et al., 2024) replaces traditional draft models with an adaptive N-gram system that updates predictions based on context. LOOKAHEAD (Fu et al., 2024) uses n-gram verification by collecting and utilizing n-grams from previous iterations as draft tokens. The N-Grammys (Stewart et al., 2024) further develops this idea by creating a dedicated n-gram based prediction system that can operate without requiring a separate draft model.

4.4 Auto-regressive Generation

Most sequence generation methods employ auto-regressive drafting, where a smaller model generates draft tokens that are verified by a larger target model. This drafting paradigm has spawned numerous techniques that vary in how the draft model interacts with the target model.

4.4.1 Independent Drafters

Auto-regressive independent drafters are techniques in which smaller model(s) generate tokens one at a time while a separate larger target model subsequently verifies the draft tokens in parallel. SpecDec (Xia et al., 2023) pioneered this approach with an independent draft model using distinct attention queries for masked positions. SpecDec++ (Huang et al., 2024) improves SpecDec (Xia et al., 2023) by training a prediction head on top of the draft model that estimates token acceptance probabilities. Based on these predictions, it dynamically determines when to stop generating tokens and trigger verification.

Recent works focus on dynamic adaptation and confidence monitoring. BiLD (Kim et al., 2024a) triggers target model verification when draft confidence falls below a threshold, while ON-THE-FLY (Liu et al., 2025b) dynamically adjusts window sizes based on prediction accuracy. OSD (Liu et al., 2023) enables online adaptation through knowledge distillation during inference, and DistillSpec (Zhou et al., 2023) extends this by accessing target model logits for improved alignment. (Liu et al., 2025a) introduces special tokens for draft models to autonomously determine target model consultation, eliminating separate verification at some performance cost. For mathematical applications, Judge(Bachmann et al., 2025) adds a learned verification layer atop the target model's embeddings, using contextual correctness assessment to reduce strict output alignment requirements.

4.4.2 Dependent Drafters

The main drawbacks of independent drafting approaches are that (1) the computation required to generate the draft tokens is fixed per tokens, meaning that computation is over-provisioned for many "easy" tokens and (2) the target model cannot reuse the features of the drafting process, increasing the amount of compute required. Self-speculative decoding approaches generate draft tokens by relying directly on a subset (**Layer Skipping**) or extension (**Dependent Heads**) of the target model.

Layer Skipping Draft&Verify (Zhang et al., 2023), SWIFT (Xia et al., 2024a), and Draft on the Fly (Metel et al., 2024) achieves fast draft token generation by selectively skipping some intermediate layers in the Draft process, and then verifies these drafts using the full LLM. In order to achieve good draft accuracy, they also designed an intermediate layer selection algorithm based on Bayesian optimization. LayerSkip (Elhoushi et al., 2024) uses an early exiting (Teerapittayanon et al., 2016) approach to dynamically output tokens at different depths of the target model. Kangaroo (Liu et al., 2024b) also applied early exit by adopting a shallow sub-network to generate drafts and using a lightweight adapter module to bridge the performance gap with the full model, achieving efficient and accurate decoding. EESD (Liu et al., 2024d) use Thompson Sampling Control (Slivkins et al., 2019) Mechanism to adaptively determines how many draft token will be generated. SPEED (Hooper et al., 2023) combines speculative execution with parameter sharing, using early predictions to process multiple tokens in parallel through shared decoder layers, rather than waiting for each token to complete sequentially.

Dependent Heads Dependent head-based drafting eliminates the need for a separate draft model by adding lightweight feed-forward prediction heads using the hidden states of the target model. The main idea is that the first token in sequence generation block uses the target model as usual but the features at the end of the model are fed into additional heads to predict subsequent tokens without passing back through the entire target model.

EAGLE (Li et al., 2024f) uses a trained head that takes in hidden states from the target model and generates subsequent draft tokens in an AR manner. Hydra (Ankner et al., 2024) use multiple decoding, one for each draft token position.

EAGLE extensions have focused on improving parallel token generation and attention mechanisms. Falcon (Gao et al., 2024) introduces a semi-autoregressive (SAR) drafting framework which combines token embeddings and transformer features from the target LLM as input, utilizing an architecture of LSTM layers, relaxed causal-masked self-attention, and MLP networks. This design allows simultaneous generation of k tokens per forward pass while remaining lightweight with just two transformer layers. HASS (Zhang et al., 2024a), improves knowledge distillation by emphasizing high-probability tokens during training. This approach ensures better alignment between training and inference, addressing the constraint that the draft model cannot access the target model's hidden states during deployment. Mixture of Attentions (Zimmer et al., 2024) further enhances EAGLE by incorporating Layer Self-Attention (LSA) for processing cross-layer hidden states, Self-Attention (SA) for tracking drafted tokens, and Cross-Attention (CA) for multi-token prediction using LSA outputs. DeepSeek-V3 (Liu et al., 2024a) adapts (Gloeckle et al., 2024)'s multi-token approach (discussed in Section 4.5) while maintaining complete causal attention during inference.

4.5 Multi-token Prediction

Stern et al. (2018) proposes adding multiple decoding heads on top of a model to predict k future tokens in parallel, requiring training the entire model from scratch. Medusa (Cai et al., 2024) introduces a parameter-efficient approach, where lightweight decoding heads are fine-tuned on top of pre-trained language models. Each head is trained to predict a specific future position in the sequence without modifying the target model. (Gloeckle et al., 2024) propose a multi-token prediction paradigm where a shared backbone optimized jointly with multiple prediction heads that enable propagation of information related to sequential tokens during training that can be discarded at inference to enable parallel generation (similar to Medusa).

Amphista (Li et al., 2024g) achieves this through bi-directional self-attention, allowing heads to consider both previous and future predictions, along with staged adaptation layers that employ two transformer decoders to bridge the gap between autoregressive and non-autoregressive architectures. Similarly, CTC-drafter (Wen et al., 2024) establishes token correlations using Connectionist Temporal Classification (CTC). The model incorporates blank tokens (ε) and allows token repetition while predicting multiple candidates for each position. A CTC transform module then processes these predictions, removing duplicates and blank tokens to produce the final draft sequences.

5 Sequence Refinement Methods

5.1 Single-pass Verification

Single-pass verification represents the most common refinement strategy in draft-and-verify approaches, where drafted tokens are verified exactly once by the target model.

5.1.1 Linear Verification

Linear verification sequentially validates draft tokens against the target model's logit distributions, with early works like SpecDec (Xia et al., 2023) and Draft&Verify (Zhang et al., 2023) comparing drafted tokens against the target model's predictions. When a token fails verification (i.e., when the draft output doesn't match the target model's distribution), the system falls back to standard AR generation from that point.

Fast Inference (Leviathan et al., 2023) and (Chen et al., 2023a) introduced speculative sampling to improve acceptance rates while approximately maintaining the target distribution. Their method accepts a token if the target model assigns equal or higher probability; otherwise, it accepts with probability $p(x)/q(x)$ or resamples from an adjusted distribution.

Block Verification (Sun et al., 2025) and MTAD (Qin et al., 2024b) improve upon linear verification by examining the joint probability distribution of draft tokens as a chain of conditional probabilities. This block-based evaluation approach typically results in higher acceptance rates compared to token-by-token verification for similar quality.

5.1.2 Tree-based Verification

Tree-based verification extends the single-pass paradigm by enabling parallel exploration of multiple completion paths. Unlike linear verification that processes a single sequence, tree-based methods construct and verify a tree of possible completions simultaneously, making more efficient use of parallel compute resources.

SpecInfer (Miao et al., 2023) pioneered this approach by developing an efficient tree-based attention masking scheme that enables parallel verification while maintaining proper token dependencies. This innovation maintains generation quality while significantly increasing the number of tokens that can be verified in parallel.

Recent works have focused on optimizing tree structure and size to maximize computational efficiency. Sequoia (Chen et al., 2024b) introduces a hardware-aware tree optimizer that can maximize inference performance by selecting appropriate tree dimensions based on available computing resources. OPT-Tree (Wang et al., 2024a) searches for optimal tree structures to maximize expected acceptance length per decoding step. DSBD (Qin et al., 2024a) uses a small model to generate multiple candidate sequences via beam search, then the

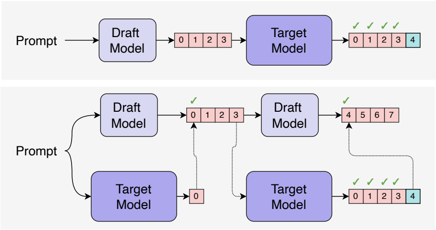


Figure 4: Comparison of speculative decoding approaches: (a) Sequential processing where draft generates tokens (0-3) before target verification. (b) Parallel processing where draft generates new tokens while target simultaneously verifies previous ones.

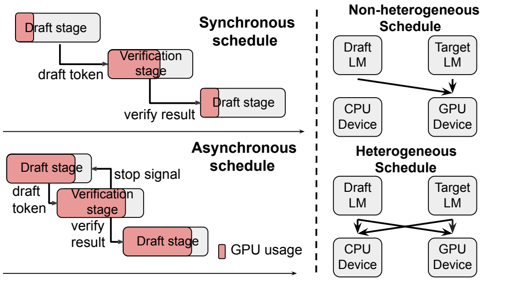
large model verifies these sequences layer by layer while dynamically adjusting the beam width based on acceptance probabilities to balance efficiency and quality. DySpec (Xiong et al., 2024) enables dynamic tree expansion during runtime based on prediction confidence, while EAGLE2 (Li et al., 2024d) incorporates context-aware tree construction to improve acceptance rates. DDD (Brown et al., 2024) optimizes EAGLE2 (Li et al., 2024d) 's tree drafting method by making the depth dynamic based on draft model confidence.

Several works have explored approaches that combine tree-based verification with other techniques. ProPD (Zhong et al., 2024) integrates progressive refinement into the tree structure, while RSD (Jeon et al., 2024) employs recursive verification. GSD (Gong et al., 2024) and ADED (Liu et al., 2024f) extend tree-based methods to capture complex dependencies through graph-based representations and adaptive depth adjustment.

In terms of verifying multiple candidate draft tokens in parallel (also known as Multi-Draft Speculative Decoding, MDSD), (Hu et al., 2025) propose a hybrid sampling strategy that combines deterministic selection of high-probability tokens with random sampling of the final token, improving acceptance rates in certain scenarios. (Khisti et al., 2024) introduce a two-phase verification method that uses importance sampling to select a draft token before applying single-draft verification, optimizing the process for parallel draft generation.

5.2 Iterative Decoding

Iterative decoding methods extend single-pass verification by allowing multiple refinement iterations on draft tokens until convergence. These approaches are inspired by classical methods for solving systems of nonlinear equations, particularly the Jacobi and Gauss-Seidel iteration methods.



(a) Asynchronous Schedule (b) Heterogeneous Schedule

Figure 5: Asynchronous and heterogeneous schedules.

In (Santilli et al., 2023), the authors reframe AR text generation as an iterative optimization problem. Their approach expresses token generation as a system where each position must output the most likely token given the current state of all other positions. Starting with a randomly initialized sequence, they adapt the Jacobi method to update all positions in parallel during each iteration until convergence. The authors prove that this process produces identical output to traditional AR decoding under greedy sampling. Fu et al. (2024) builds upon this framework with LOOKAHEAD decoding, which combines Jacobi iterations with n-gram verification to accelerate convergence by leveraging predictions from earlier steps.

CLLMs (Kou et al., 2024) leverages consistency training to accelerate convergence by enabling better multi-token prediction in early iterations.

6 System-Level Optimizations and Implementation Strategies

6.1 Parallel Speculative Decoding

Traditional SD processes tokens sequentially, with the draft model generating tokens followed by target model verification, creating inherent bottlenecks. As shown in Figure 4, parallel approaches overcome this limitation by enabling simultaneous operation - while the target model verifies earlier tokens, the draft model generates subsequent ones, enabling continuous overlapped execution. Recent methods build upon this paradigm: CS Drafting (Chen et al., 2023b) employs vertical and horizontal cascade structures for 81% speedup, PaSS (Monea et al., 2023) uses look-ahead embeddings for 30% speedup, and Faster Cascades (Narasimhan et al., 2024) incorporates deferral rules for improved cost-quality trade-offs. PEARL (Liu et al., 2024e) further advances this through pre-verify and post-verify strategies with adaptive draft lengths, achieving

4.43× speedup over AR decoding and 1.50× over standard SD AMUSD (McDanel, 2024) presents an asynchronous multi-device approach to SD, decoupling the draft and verify phases into continuous, asynchronous operations.

6.2 Distributed Speculative Decoding

Edge computing environments impose stringent constraints on memory, compute power, and latency, necessitating specialized SD approaches to deploy LLMs effectively in resource-constrained settings. SpecExec (Svirschevski et al., 2024) is designed to harness the parallel processing power of consumer GPUs to accelerate LLM inference. By generating multiple tokens per target model iteration and constructing a "cache" tree of probable continuations, SpecExec efficiently validates these continuations with the target model in a single pass. EdgeLLM (Xu et al., 2024a) further optimizes on-device LLM inference through novel techniques for resource allocation and error correction, achieving great token generation speeds and significantly outperforming existing engines. Dovetail (Zhang et al., 2024b) represents a significant advancement in heterogeneous computing for LLM inference. By deploying the draft model on the GPU and the target model on the CPU, Dovetail reduces the granularity of data transfer and enhances the overall inference process. The introduction of Dynamic Gating Fusion (DGF) and optimizations for low-end hardware further improve the balance between latency and performance.

6.3 Compiler and Hardware Optimization for Speculative Decoding

Efficient implementation of SD requires careful optimization of both hardware resources and compiler strategies to maximize throughput and minimize latency. SpecPIM (Li et al., 2024b) presents a novel approach to accelerate speculative inference on a Processing-in-Memory (PIM) system through co-exploration of architecture and dataflow. This method constructs a design space that comprehensively considers algorithmic and architectural heterogeneity, enabling optimal hardware resource allocation for different models and computational patterns. (Wagner et al., 2024) investigates improvements in speculative sampling on GPUs, achieving significant speed gains by parallelizing computations and using sigmoid approximations for softmax, though this comes with a minor reduction in accuracy.

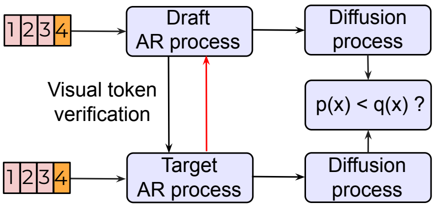


Figure 6: Flow of AR image generation with SD.

Recent studies have focused on enhancing the throughput of LLMs using SD by optimizing batch processing and scheduling strategies. Figure 5 illustrates two scheduling strategies for SD systems: (a) Asynchronous Schedule: The draft stage is followed by the verify stage, with optional stop signals determining further processing. This non-blocking approach enhances system efficiency. (b) Heterogeneous Schedule: Both CPU and GPU devices are utilized for different stages of the decoding process, enabling parallel processing and optimizing performance through resource allocation. Using Markov chain theory, (Yin et al., 2024) establishes SD's optimality among unbiased algorithms while highlighting the tradeoff between inference speed and output quality. Their analysis reveals that batch processing benefits are limited by the distribution gap between small and large models. MagicDec (Chen et al., 2024a) identifies the shift from compute-bound to memory-bound bottlenecks as batch size and sequence length increase, using sparse KV caches in draft models to optimize throughput. BASS (Qian et al., 2024) extends SD to a batched setting with customized CUDA kernels for ragged tensors in attention calculations and dynamically adjusts draft lengths for better GPU utilization. SEED (Wang et al., 2024c) accelerates reasoning tree construction through scheduled speculative execution, using a rounds-scheduled strategy for conflict-free parallel processing. PipeInfer (Butler et al., 2024) addresses single-request latency through pipelined speculative acceleration, reducing inter-token latency via asynchronous speculation and early cancellation. TRIFORCE (Sun et al., 2024a) introduces a hierarchical SD mechanism with a dynamic sparse KV cache to achieve lossless acceleration of long sequence generation, significantly improving generation speed and efficiency while maintaining quality. (Zhao et al., 2024a) proposes QSPEC, a novel framework that combines weight-shared quantization schemes with SD, achieving up to 1.55× acceleration without quality loss, paving the way for efficient and highfidelity quantization deployment in diverse and memory-constrained settings. (Wang et al., 2024b) introduces a hardware-aware SD algorithm that accelerates the inference speed of Mamba and hybrid models. Inspired by SD, SKD (Xu et al., 2024b) represents a novel, adaptive approach to knowledge distillation. By dynamically generating tokens and using the teacher model to filter or replace low-quality samples, it bridges the gap between supervised KD's reliance on static data and on-policy KD's susceptibility to low-quality outputs. This ensures a better alignment between training and inference distributions, and improved performance.

7 Multimodal Models and Applications

7.1 Speculative Decoding for Visual Output Generation

Researchers are now using SD to improve the efficiency of AR image generation (Ding et al., 2021; Yu et al., 2022; Li et al., 2024c). As shown in Figure 6, this method greatly speeds up the process by reducing the inference steps needed for generating visual tokens. For instance, (Wang et al., 2024d) proposes a novel continuous SD method that designs a novel acceptance criterion for the diffusion distributions, significantly improving the efficiency of AR image generation. Similarly, LANTERN (Jang et al., 2024) presents a relaxed acceptance condition for the SD strategy to substantially speed up the inference process in visual AR models. Additionally, Speculative Jacobi Decoding (SJD) (Teng et al., 2024) offers a training-free speculative Jacobi decoding technique that effectively accelerates text-to-image generation tasks.

7.2 Speculative Decoding for Multimodal Output Generation

Recent advancements in SD have substantially improve the efficiency and quality of AR generation across various modalities. In the domain of speech synthesis, VADUSA (Li et al., 2024a) leverages SD to accelerate the inference process in AR text-to-speech (TTS) systems, which enhances the quality speech synthesis as well. Inspired by the flavor of SD, (Raj et al., 2024) introduces a multi-token prediction mechanism, offering substantial improvements in inference efficiency for speech generation.

In the context of multimodal large language models, (Gagrani et al., 2024) investigates the integration of SD into the LLaVA 7B model to optimize inference efficiency. Their findings indicate that employing a lightweight, language-only draft model facilitates a memory-constrained acceleration of up to 2.37×. Besides, IbED (Lee et al.) proposes the "In-batch Ensemble Drafting" method to further enhance the robustness and efficiency of SD. It adopts the ensemble techniques during batch-level inference, requires no additional model parameters and significantly increases the validation probability of draft tokens, thereby improving performance and robustness across diverse input scenarios.

7.3 Recommendation Systems

LLM-based recommendation systems have shown great potential in enhancing personalized recommendations, but their high inference latency poses a significant challenge for real-world deployment. To address this, recent research has focused on optimizing decoding efficiency to accelerate recommendation generation. (Xi et al., 2024) propose DARE that integrates retrieval-based SD to accelerate recommendation knowledge generation, thereby improving the deployment efficiency of LLM-based recommender systems in industrial settings. AtSpeed (Lin et al., 2024) combines strict top-K alignment (AtSpeed-S) and relaxed sampling verification (AtSpeed-R), to significantly accelerate LLM-based generative recommendation with speedup from 2× to 2.5×, addressing inference latency challenges in top-K sequence generation.

8 Conclusion

This survey analyzes generation-refinement frameworks for mitigating sequential dependencies in autoregressive models, highlighting how these approaches are fundamentally changing efficient neural sequence generation across text, speech, and visual domains. Through examining both algorithmic innovations and system-level implementations, we have demonstrated their broad applicability while providing crucial deployment insights for practitioners. Moving forward, significant challenges persist in constructing solid theoretical foundations to grasp the balance between parallelism and quality, as well as in developing comprehensive approaches that span different modalities—efforts that could narrow the divide between the capabilities of large models and their actual implementation. Additionally, it remains crucial to examine the scalability of the speculative decoding system as the quantity of draft and target models increases.

Limitations

While this survey provides a comprehensive overview of generation-refinement frameworks, some limitations should be acknowledged. Detailed performance comparisons across different approaches are challenging due to varying experimental settings, model architectures, and hardware configurations used in the original papers. The lack of standardized benchmarks for speculative decoding makes it difficult to make definitive claims about the relative efficiency of different methods. Additionally, while we examine applications across different modalities, our analysis may not fully capture all domain-specific challenges and optimizations, particularly for emerging areas like video generation and multimodal reasoning.

References

2024. Open-sora report v1.1. https://github.com/hpcaitech/Open-Sora/blob/main/docs/report_02.md.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. 2024. Hydra: Sequentially-dependent draft heads for medusa decoding. *arXiv preprint arXiv:2402.05109*.

Gregor Bachmann, Sotiris Anagnostidis, Albert Pumarola, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Edgar Schönfeld, Ali Thabet, and Jonas K Kohler. 2025. Judge decoding: Faster speculative sampling requires going beyond model alignment. In *The Thirteenth International Conference on Learning Representations*.

Sangmin Bae, Jongwoo Ko, Hwanjun Song, and Se-Young Yun. 2023. Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding. *arXiv preprint arXiv:2310.05424*.

Nikhil Bhendawade, Irina Belousova, Qichen Fu, Henry Mason, Mohammad Rastegari, and Mahyar Najibi. 2024. Speculative streaming: Fast llm inference without auxiliary models. *arXiv preprint arXiv:2402.11131*.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Oscar Brown, Zhengjie Wang, Andrea Do, Nikhil Mathew, and Cheng Yu. 2024. Dynamic depth decoding: Faster speculative decoding for llms. *arXiv preprint arXiv:2409.00142*.

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. *Advances in neural information processing systems*, 33:1877–1901.

Branden Butler, Sixing Yu, Arya Mazaheri, and Ali Jannesari. 2024. Pipeinfer: Accelerating llm inference using asynchronous pipelined speculation. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–19. IEEE.

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. 2023a. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Hongruixuan Chen, Jian Song, Olivier Dietrich, Clifford Broni-Bediako, Weihao Xuan, Junjue Wang, Xinlei Shao, Yimin Wei, Junshi Xia, Cuiling Lan, Konrad Schindler, and Naoto Yokoya. 2025. Bright: A globally distributed multimodal building damage assessment dataset with very-high-resolution for all-weather disaster response. *arXiv preprint arXiv:2501.06019*.

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*.

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. 2024b. Sequoia: Scalable, robust, and hardware-aware speculative decoding. *arXiv preprint arXiv:2402.12374*.

Ziyi Chen, Xiaocong Yang, Jiacheng Lin, Chenkai Sun, Kevin Chen-Chuan Chang, and Jie Huang. 2023b. Cascade speculative drafting for even faster llm inference. *arXiv preprint arXiv:2312.11462*.

Yunfei Cheng, Aonan Zhang, Xuanyu Zhang, Chong Wang, and Yi Wang. 2024. Recurrent drafter for fast speculative decoding in large language models. *arXiv preprint arXiv:2403.09919*.

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

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. 2021. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems*, 34:19822–19835.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. 2024. Layer skip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2402.02057*.

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*.

Xiangxiang Gao, Weisheng Xie, Yiwei Xiang, and Feng Ji. 2024. Falcon: Faster and parallel inference of large language models through enhanced semi-autoregressive drafting and custom-designed decoding tree. *arXiv preprint arXiv:2412.12639*.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*.

Zhuocheng Gong, Jiahao Liu, Ziyue Wang, Pengfei Wu, Jingang Wang, Xunliang Cai, Dongyan Zhao, and Rui Yan. 2024. Graph-structured speculative decoding. *arXiv preprint arXiv:2407.16207*.

Muhammad Usman Hadi, R Qureshi, A Shah, M Irfan, A Zafar, MB Shaikh, N Akhtar, J Wu, and S Mirjalili. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *TechRxiv*.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D Lee, and Di He. 2023. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Hasan Genc, Kurt Keutzer, Amir Gholami, and Sophia Shao. 2023. Speed: Speculative pipelined execution for efficient decoding. *arXiv preprint arXiv:2310.12072*.

Zhengmian Hu, Tong Zheng, Vignesh Viswanathan, Ziyi Chen, Ryan A. Rossi, Yihan Wu, Dinesh Manocha, and Heng Huang. 2025. Towards optimal multi-draft speculative decoding. In *The Thirteenth International Conference on Learning Representations*.

Hui Huang, Shuangzhi Wu, Xinnian Liang, Bing Wang, Yanrui Shi, Peihao Wu, Muyun Yang, and Tiejun Zhao. 2023. Towards making the most of llm for translation quality estimation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 375–386. Springer.

Kaixuan Huang, Xudong Guo, and Mengdi Wang. 2024. Specdec++: Boosting speculative decoding via adaptive candidate lengths. *arXiv preprint arXiv:2405.19715*.

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*.

Wonseok Jeon, Mukul Gagrani, Raghavv Goel, Junyoung Park, Mingu Lee, and Christopher Lott. 2024. Recursive speculative decoding: Accelerating llm inference via sampling without replacement. *arXiv preprint arXiv:2402.14160*.

Ashish Khisti, M Reza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. 2024. Multi-draft speculative sampling: Canonical architectures and theoretical limits. *arXiv preprint arXiv:2410.18234*.

Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W Mahoney, Amir Gholami, and Kurt Keutzer. 2024a. Speculative decoding with big little decoder. *Advances in Neural Information Processing Systems*, 36.

Taehyeon Kim, Ananda Theertha Suresh, Kishore A Papineni, Michael Riley, Sanjiv Kumar, and Adrian Benton. 2024b. Accelerating blockwise parallel language models with draft refinement. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. 2024. Cllms: Consistency large language models. *arXiv preprint arXiv:2403.00835*.

Minjae Lee, Wonjun Kang, Minghao Yan, Christian Classen, Hyung Il Koo, and Kangwook Lee. In-batch ensemble drafting: Toward fast and robust speculative decoding for multimodal language models.

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.

Bohan Li, Hankun Wang, Situo Zhang, Yiwei Guo, and Kai Yu. 2024a. Fast and high-quality auto-regressive speech synthesis via speculative decoding. *arXiv preprint arXiv:2410.21951*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2023. Seed-bench-2: Benchmarking multimodal large language models. *arXiv preprint arXiv:2311.17092*.

Cong Li, Zhe Zhou, Size Zheng, Jiaxi Zhang, Yun Liang, and Guangyu Sun. 2024b. Specpim: Accelerating speculative inference on pim-enabled system via architecture-dataflow co-exploration. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 950–965.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. 2024c. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024d. Eagle-2: Faster inference of language models with dynamic draft trees. *Preprint*, arXiv:2406.16858.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024e. Eagle-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*.

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

Zeping Li, Xinlong Yang, Ziheng Gao, Ji Liu, Zhuang Liu, Dong Li, Jinzhang Peng, Lu Tian, and Emad Barsoum. 2024g. Amphista: Accelerate llm inference with bi-directional multiple drafting heads in a non-autoregressive style. *arXiv preprint arXiv:2406.13170*.

Xinyu Lin, Chaoqun Yang, Wenjie Wang, Yongqi Li, Cunxiao Du, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Efficient inference for large language model-based generative recommendation. *arXiv preprint arXiv:2410.05165*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Fangcheng Liu, Yehui Tang, Zhenhua Liu, Yunsheng Ni, Kai Han, and Yunhe Wang. 2024b. Kangaroo: Lossless self-speculative decoding via double early exiting. *arXiv preprint arXiv:2404.18911*.

Guanlin Liu, Anand Ramachandran, Tanmay Gangwani, Yan Fu, and Abhinav Sethy. 2025a. Knowledge distillation with training wheels.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024c. Llava-next: Improved reasoning, ocr, and world knowledge.

Jiahao Liu, Qifan Wang, Jingang Wang, and Xunliang Cai. 2024d. Speculative decoding via early-exiting for faster llm inference with thompson sampling control mechanism. *arXiv preprint arXiv:2406.03853*.

Jiesong Liu, Brian Park, and Xipeng Shen. 2025b. A drop-in solution for on-the-fly adaptation of speculative decoding in large language models.

Tianyu Liu, Yun Li, Qitan Lv, Kai Liu, Jianchen Zhu, and Winston Hu. 2024e. Parallel speculative decoding with adaptive draft length. *arXiv preprint arXiv:2408.11850*.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. 2023. Online speculative decoding. *arXiv preprint arXiv:2310.07177*.

Xukun Liu, Bowen Lei, Ruqi Zhang, and Dongkuan Xu. 2024f. Adaptive draft-verification for efficient large language model decoding. *arXiv preprint arXiv:2407.12021*.

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

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. 2025. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*.

Ahmed Masry, Do Long, 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, Dublin, Ireland. Association for Computational Linguistics.

Bradley McDanel. 2024. Amusd: Asynchronous multi-device speculative decoding for llm acceleration. *arXiv preprint arXiv:2410.17375*.

Bradley McDanel, Sai Qian Zhang, Yunhai Hu, and Zining Liu. 2025. Pipespec: Breaking stage dependencies in hierarchical llm decoding. *Preprint*, arXiv:2505.01572.

Michael R Metel, Peng Lu, Boxing Chen, Mehdi Rezagholizadeh, and Ivan Kobyzev. 2024. Draft on the fly: Adaptive self-speculative decoding using cosine similarity. *arXiv preprint arXiv:2410.01028*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. 2023. Specinfer: Accelerating generative large language model serving with tree-based speculative inference and verification. *arXiv preprint arXiv:2305.09781*.

Giovanni Monea, Armand Joulin, and Edouard Grave. 2023. Pass: Parallel speculative sampling. *arXiv preprint arXiv:2311.13581*.

Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Faster cascades via speculative decoding. *arXiv preprint arXiv:2405.19261*.

Jie Ou, Yueming Chen, and Wenhong Tian. 2024. Lossless acceleration of large language model via adaptive n-gram parallel decoding. *arXiv preprint arXiv:2404.08698*.

Haifeng Qian, Sujan Kumar Gonugondla, Sungsoo Ha, Mingyue Shang, Sanjay Krishna Gouda, Ramesh Nallapati, Sudipta Sengupta, Xiaofei Ma, and Anoop Deoras. 2024. Bass: Batched attention-optimized speculative sampling. *arXiv preprint arXiv:2404.15778*.

Zongyue Qin, Zifan He, Neha Prakriya, Jason Cong, and Yizhou Sun. 2024a. Dynamic-width speculative beam decoding for efficient llm inference. *arXiv preprint arXiv:2409.16560*.

Zongyue Qin, Ziniu Hu, Zifan He, Neha Prakriya, Jason Cong, and Yizhou Sun. 2024b. Optimized multi-token joint decoding with auxiliary model for llm inference. *arXiv preprint arXiv:2407.09722*.

Desh Raj, Gil Keren, Junteng Jia, Jay Mahadeokar, and Ozlem Kalinli. 2024. Faster speech-llama inference with multi-token prediction. *arXiv preprint arXiv:2409.08148*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodolà. 2023. Accelerating transformer inference for translation via parallel decoding. *arXiv preprint arXiv:2305.10427*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286.

Benjamin Spector and Chris Re. 2023. Accelerating llm inference with staged speculative decoding. *arXiv preprint arXiv:2308.04623*.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31.

Lawrence Stewart, Matthew Trager, Sujan Kumar Gonugondla, and Stefano Soatto. 2024. The n-grammys: Accelerating autoregressive inference with learning-free batched speculation. *arXiv preprint arXiv:2411.03786*.

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*.

Ryan Sun, Tianyi Zhou, Xun Chen, and Lichao Sun. 2024b. Spechub: Provable acceleration to multi-draft speculative decoding. *arXiv preprint arXiv:2411.05289*.

Ziteng Sun, Uri Mendlovic, Yaniv Leviathan, Asaf Aharoni, Ahmad Beirami, Jae Hun Ro, and Ananda Theertha Suresh. 2025. Block verification accelerates speculative decoding. In *The Thirteenth International Conference on Learning Representations*.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. 2024c. Spectr: Fast speculative decoding via optimal transport. *Advances in Neural Information Processing Systems*, 36.

Ruslan Svirschevski, Avner May, Zhuoming Chen, Beidi Chen, Zhihao Jia, and Max Ryabinin. 2024. Specexec: Massively parallel speculative decoding for interactive llm inference on consumer devices. *arXiv preprint arXiv:2406.02532*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Surat Teerapittayanon, Bradley McDanel, and H.T. Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE.

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*.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*.

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*.

Dominik Wagner, Seanie Lee, Ilja Baumann, Philipp Seeberger, Korbinian Riedhammer, and Tobias Bocklet. 2024. Optimized speculative sampling for gpu hardware accelerators. *arXiv preprint arXiv:2406.11016*.

Jikai Wang, Yi Su, Juntao Li, Qingrong Xia, Zi Ye, Xinyu Duan, Zhefeng Wang, and Min Zhang. 2024a. Opt-tree: Speculative decoding with adaptive draft tree structure. *arXiv preprint arXiv:2406.17276*.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M Rush, and Tri Dao. 2024b. The mamba in the llama: Distilling and accelerating hybrid models. *arXiv preprint arXiv:2408.15237*.

Zhenglin Wang, Jialong Wu, Yilong Lai, Congzhi Zhang, and Deyu Zhou. 2024c. Seed: Accelerating reasoning tree construction via scheduled speculative decoding. *arXiv preprint arXiv:2406.18200*.

Zili Wang, Robert Zhang, Kun Ding, Qi Yang, Fei Li, and Shiming Xiang. 2024d. Continuous speculative decoding for autoregressive image generation. *arXiv preprint arXiv:2411.11925*.

Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, et al. 2024e. Speculative rag: Enhancing retrieval augmented generation through drafting. *arXiv preprint arXiv:2407.08223*.

Zhuofan Wen, Shangtong Gui, and Yang Feng. 2024. Speculative decoding with ctc-based draft model for llm inference acceleration. *arXiv preprint arXiv:2412.00061*.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633.

Yunjia Xi, Hangyu Wang, Bo Chen, Jianghao Lin, Menghui Zhu, Weiwen Liu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. A decoding acceleration framework for industrial deployable llm-based recommender systems. *arXiv preprint arXiv:2408.05676*.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2023. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3909–3925.

Heming Xia, Yongqi Li, Jun Zhang, Cunxiao Du, and Wenjie Li. 2024a. Swift: On-the-fly self-speculative decoding for llm inference acceleration. *arXiv preprint arXiv:2410.06916*.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024b. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7655–7671, Bangkok, Thailand. Association for Computational Linguistics.

Zilin Xiao, Hongming Zhang, Tao Ge, Siru Ouyang, Vicente Ordonez, and Dong Yu. 2024. Parallelspec: Parallel drafter for efficient speculative decoding. *arXiv preprint arXiv:2410.05589*.

Yunfan Xiong, Ruoyu Zhang, Yanzeng Li, Tianhao Wu, and Lei Zou. 2024. Dyspec: Faster speculative decoding with dynamic token tree structure. *arXiv preprint arXiv:2410.11744*.

Daliang Xu, Wangsong Yin, Hao Zhang, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and Xuanzhe Liu. 2024a. Edgellm: Fast on-device llm inference with speculative decoding. *IEEE Transactions on Mobile Computing*.

Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. 2024b. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023a. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39.

Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Inference with reference: Lossless acceleration of large language models. *arXiv preprint arXiv:2304.04487*.

Seongjun Yang, Gibbeum Lee, Jaewoong Cho, Dimitris Papailiopoulos, and Kangwook Lee. 2023c. Predictive pipelined decoding: A compute-latency trade-off for exact llm decoding. *arXiv preprint arXiv:2307.05908*.

Hanling Yi, Feng Lin, Hongbin Li, Ning Peiyang, Xiaotian Yu, and Rong Xiao. 2024. Generation meets verification: Accelerating large language model inference with smart parallel auto-correct decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5285–5299, Bangkok, Thailand. Association for Computational Linguistics.

Ming Yin, Minshuo Chen, Kaixuan Huang, and Mengdi Wang. 2024. A theoretical perspective for speculative decoding algorithm. *arXiv preprint arXiv:2411.00841*.

Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, Jiayi Lei, Quanfeng Lu, Runjian Chen, Peng Xu, Renrui Zhang, Haozhe Zhang, Peng Gao, Yali Wang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. 2024. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. *Preprint*, arXiv:2404.16006.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5.

Ofir Zafrir, Igor Margulis, Dorin Shteyman, and Guy Boudoukh. 2024. Fastdraft: How to train your draft. *arXiv preprint arXiv:2411.11055*.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2023. Draft & verify: Lossless large language model acceleration via self-speculative decoding. *arXiv preprint arXiv:2309.08168*.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. 2024a. Learning harmonized representations for speculative sampling. *arXiv preprint arXiv:2408.15766*.

Libo Zhang, Zhaoning Zhang, Baizhou Xu, Songzhu Mei, and Dongsheng Li. 2024b. Dovetail: A cpu/gpu heterogeneous speculative decoding for llm inference. *arXiv preprint arXiv:2412.18934*.

Juntao Zhao, Wenhao Lu, Sheng Wang, Lingpeng Kong, and Chuan Wu. 2024a. Qspec: Speculative decoding with complementary quantization schemes. *arXiv preprint arXiv:2410.11305*.

Weilin Zhao, Yuxiang Huang, Xu Han, Wang Xu, Chaojun Xiao, Xinrong Zhang, Yewei Fang, Kaihuo Zhang, Zhiyuan Liu, and Maosong Sun. 2024b. Ouroboros: Generating longer drafts phrase by phrase for faster speculative decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13378–13393, Miami, Florida, USA. Association for Computational Linguistics.

Shuzhang Zhong, Zebin Yang, Meng Li, Ruihao Gong, Runsheng Wang, and Ru Huang. 2024. Propd: Dynamic token tree pruning and generation for llm parallel decoding. *arXiv preprint arXiv:2402.13485*.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2023. Distillspec: Improving speculative decoding via knowledge distillation. *arXiv preprint arXiv:2310.08461*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. Toolqa: A dataset for llm question answering with external tools. *arXiv preprint arXiv:2306.13304*.

Matthieu Zimmer, Milan Gritta, Gerasimos Lampouras, Haitham Bou Ammar, and Jun Wang. 2024. Mixture of attentions for speculative decoding. *arXiv preprint arXiv:2410.03804*.

A Performance Evaluation

Experimental Setup

We follow the public implementation of (Xia et al., 2024b) and run all experiments on a single NVIDIA A100 80GB GPU with a batch size of 1. We evaluate different speculative decoding (SD) methods across both language-only model (LLMs) and vision-language models (VLMs), and report two primary metrics:

Speed-up Ratio (S): defined as $t_{\text{AR}}/t_{\text{method}}$, where t_{AR} is the average wall-clock time per token under standard autoregressive decoding and t_{method} is the same under the evaluated method.

Average Accepted Token Length (τ): the number of consecutive draft tokens accepted per verification step, reflecting decoding efficiency and draft model quality.

For LLM evaluation, we benchmark six SD methods on standard language tasks: MT-Bench (Ying et al., 2024) (conversation), WMT16 EN-RO (Bojar et al., 2016) (translation), CNN/DM (See et al., 2017) (summarization), GPQA (Rein et al.) and GSM8K (Cobbe et al., 2021) (question answering), and BRIGHT (Chen et al., 2025) (retrieval-augmented generation).

For VLM evaluation, we test six SD baselines—SPD (Gagrani et al., 2024), Kangaroo (Liu et al., 2024b), Medusa (Cai et al., 2024), Hydra (Ankner et al., 2024), EAGLE (Li et al., 2024f), and EAGLE2 (Li et al., 2024e)—on five models of varying sizes: LLaVA-v1.6-Vicuna (7B, 13B)(Liu et al., 2024c), Pixtral (12B)(Agrawal et al., 2024), SmolVLM (2B)(Marafioti et al., 2025), and Gemma3 (12B)(Team et al., 2025). Evaluations are conducted across eight benchmarks: MMT-Bench (Ying et al., 2024), SEED-Bench-2 (Li et al., 2023), ScienceQA (Lu et al., 2022), OCRBench (Liu et al., 2024g), ChartQA (Masry et al., 2022), and MathVista (Lu et al., 2024).

Model	**MT-bench**	**WMT16 EN-RO**	**CNN/DM**	**GPQA**	**GSM8K**	**BRIGHT**	**#Acc. Tok.**	**Overall**
	(conv.)	(trans.)	(summ.)	(QA)	(QA)	(RAG)	(τ)	(S)
EAGLE2	2.58 ×	1.80 ×	2.11 ×	2.06 ×	2.72 ×	1.85 ×	4.39	2.17 ×
HASS	2.43 ×	1.75 ×	2.05 ×	1.99 ×	2.52 ×	1.80 ×	3.96	2.11 ×
EAGLE	2.26 ×	1.68 ×	2.07 ×	1.93 ×	2.31 ×	1.81 ×	3.56	1.98 ×
Hydra	2.16 ×	1.82 ×	1.69 ×	1.84 ×	2.23 ×	1.67 ×	3.33	1.96 ×
SpS	1.98 ×	1.32 ×	2.01 ×	1.85 ×	1.88 ×	1.78 ×	2.20	1.87 ×
PLD	1.69 ×	1.22 ×	2.41 ×	1.27 ×	1.65 ×	1.74 ×	1.78	1.60 ×
Recycling	1.47 ×	1.38 ×	1.40 ×	1.25 ×	1.66 ×	1.37 ×	2.64	1.48 ×
Kangaroo	1.63 ×	1.24 ×	1.50 ×	1.43 ×	1.61 ×	1.52 ×	2.31	1.48 ×
Medusa	1.58 ×	1.42 ×	1.23 ×	1.48 ×	1.70 ×	1.17 ×	2.37	1.46 ×
REST	1.41 ×	1.22 ×	1.10 ×	1.32 ×	1.35 ×	1.20 ×	1.71	1.24 ×
Lookahead	1.13 ×	1.03 ×	1.17 ×	1.07 ×	1.24 ×	1.10 ×	1.63	1.12 ×

Table 1: Speed-up over greedy autoregressive decoding on six benchmarks. "#Acc. Tok." counts mean accepted tokens per verification step; "Overall" is the geometric mean across tasks.

		MMT		SEED		ScienceQA		OCRBench		ChartQA		MathVista		Average	
Models	Methods	S	τ	S	τ	S	τ	S	τ	S	τ	S	τ	S	τ
Temperature = 0															
LLaVA-v1.6Vicuna-7B	SPD (Gagrani et al., 2024)	1.10	1.88	0.81	1.17	1.08	1.87	0.89	1.25	0.91	1.24	1.06	1.76	0.97	1.53
	Kangaroo (Liu et al., 2024b)	1.32	2.11	1.33	2.12	1.31	2.09	1.17	1.89	1.18	1.98	1.15	1.86	1.24	2.01
	Medusa (Cai et al., 2024)	1.58	2.88	1.59	3.01	1.44	2.77	1.22	2.33	1.25	2.41	1.22	2.34	1.38	2.62
	Hydra (Ankner et al., 2024)	1.78	3.86	1.72	3.88	1.68	3.79	1.41	3.21	1.35	3.11	1.42	3.25	1.56	3.52
	EAGLE (Li et al., 2024f)	2.10	5.04	2.09	5.01	1.98	4.88	1.72	4.13	1.56	3.98	1.78	4.25	1.87	4.55
	EAGLE-2 (Li et al., 2024e)	2.31	5.48	2.31	5.61	2.15	5.22	1.92	4.88	1.77	4.22	1.87	4.67	2.05	5.01
LLaVA-v1.6Vicuna-13B	SPD	1.07	1.78	1.06	1.79	1.09	1.88	0.86	1.12	0.89	1.25	0.87	1.22	1.00	1.58
	Kangaroo	1.43	1.77	1.51	1.87	1.22	1.55	1.21	1.54	1.27	1.61	1.53	2.01	1.36	1.72
	Medusa	1.99	2.67	1.96	2.76	1.93	2.77	1.40	2.92	1.51	2.82	1.51	2.62	1.72	2.76
	Hydra	2.12	2.87	2.08	2.99	2.21	3.12	1.49	3.07	1.65	3.03	1.66	2.87	1.87	2.99
	EAGLE	2.45	3.56	2.19	3.24	2.63	3.98	1.65	3.31	1.85	3.27	1.8	3.09	2.10	3.41
	EAGLE-2	2.89	4.05	3.18	4.33	3.09	4.97	2.20	4.12	2.41	4.15	2.39	3.76	2.69	4.23
Pixtral-12B	SPD	1.08	1.51	1.03	1.47	1.05	1.49	1.05	1.49	1.04	1.43	1.04	1.46	1.05	1.47
	Kangaroo	1.26	1.54	1.09	1.39	1.14	1.51	1.16	1.52	1.12	1.47	1.13	1.49	1.15	1.49
	Medusa	1.37	1.81	1.37	1.81	1.35	1.87	1.24	1.69	1.22	1.68	1.16	1.47	1.28	1.72
	Hydra	1.58	2.24	1.47	2.04	1.53	2.06	1.38	1.81	1.34	1.79	1.36	1.78	1.44	1.95
	EAGLE	2.38	3.47	1.97	2.53	2.31	3.64	1.69	2.73	1.78	2.84	1.64	2.47	1.96	2.95
	EAGLE-2	2.81	3.95	2.31	3.07	2.64	4.03	2.12	3.25	2.14	3.17	1.81	2.73	2.31	3.37
SmolVLM-2B	SPD	1.02	1.33	1.04	1.41	1.06	1.43	1.06	1.42	1.07	1.46	1.02	1.34	1.04	1.40
	Kangaroo	1.28	1.48	1.08	1.18	1.03	1.17	1.06	1.22	1.04	1.14	1.08	1.23	1.10	1.24
	Medusa	2.12	2.71	1.51	2.00	1.72	2.22	1.20	1.61	1.15	1.55	1.35	1.75	1.51	1.97
	Hydra	2.33	3.07	1.62	2.08	1.98	2.66	1.32	1.74	1.22	1.58	1.51	1.98	1.66	2.19
	EAGLE	2.57	3.42	1.85	2.56	2.16	2.76	1.42	1.88	1.34	1.77	1.65	2.22	1.83	2.44
	EAGLE-2	2.96	3.89	2.12	2.93	2.39	3.21	1.65	2.11	1.51	2.13	1.81	2.63	2.07	2.82
Gemma3-12B	Kangaroo	1.37	1.66	1.47	1.79	1.52	1.57	3.17	2.28	2.28	1.85	1.18	1.64	1.83	1.80
	EAGLE	1.73	1.98	1.69	2.52	1.72	1.97	4.26	2.42	3.40	1.99	1.42	1.89	2.37	2.13
	EAGLE-2	2.92	1.99	1.74	2.79	1.92	1.98	4.68	2.57	3.48	2.23	1.52	1.91	2.71	2.25
Temperature = 1															
LLaVA-v1.6Vicuna-7B	SPD	0.83	1.19	0.81	1.15	0.85	1.18	0.75	1.06	0.72	1.08	0.92	1.48	0.81	1.19
	Kangaroo	1.20	1.97	1.26	2.03	1.23	2.01	1.09	1.80	1.11	1.89	1.07	1.77	1.16	1.91
	EAGLE-2	2.19	5.37	2.20	5.48	2.04	5.12	1.82	4.77	1.68	4.13	1.76	4.56	1.95	4.91
LLAVA-v1.6Vicuna-13B	SPD	0.88	1.22	0.84	1.25	0.84	1.32	0.79	1.18	0.81	1.14	0.88	1.24	0.84	1.22
	Kangaroo	1.23	1.57	1.17	1.53	1.07	1.44	1.01	1.24	1.07	1.34	1.21	1.67	1.13	1.46
	EAGLE-2	2.35	3.75	3.02	4.30	3.03	4.67	2.03	3.87	2.18	3.83	2.18	3.41	2.46	3.97
Pixtral-12B	SPD	0.81	1.15	0.79	1.11	0.80	1.12	0.80	1.13	0.75	1.07	0.77	1.09	0.79	1.11
	Kangaroo	1.18	1.41	1.08	1.35	1.03	1.36	1.19	1.48	1.14	1.45	1.09	1.41	1.12	1.41
	EAGLE-2	2.76	3.81	2.24	3.01	2.76	3.87	2.23	3.24	2.03	3.09	1.79	2.69	2.30	3.28
SmolVLM-2B	SPD	1.07	1.47	1.01	1.33	1.07	1.46	0.97	1.26	1.06	1.44	0.85	1.20	1.00	1.36
	Kangaroo	1.37	1.59	1.12	1.24	1.22	1.41	1.12	1.29	1.18	1.36	1.28	1.42	1.22	1.39
	EAGLE-2	2.62	3.60	1.92	2.67	2.24	3.11	1.41	1.77	1.60	2.18	1.77	2.49	1.93	2.64
Gemma3-12B	Kangaroo	1.83	1.66	1.23	2.61	1.56	2.29	3.34	2.27	2.23	1.86	1.16	1.65	1.89	2.06
	EAGLE	2.23	1.96	1.60	2.52	2.16	1.97	3.74	2.65	3.30	2.03	1.59	1.86	2.44	2.16
	EAGLE-2	2.73	1.94	2.13	2.79	2.21	2.07	4.67	2.47	3.35	2.23	1.65	1.89	2.79	2.23

Table 2: Accuracy evaluation results in speedup ratios (S) and average accepted token length (τ).