ASYMMETRIC SPECULATIVE DECODING: UNLOCKING IN-CONTEXT LEARNING IN DRAFT MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding addresses Large Language Model (LLM) inference latency by first drafting and then verifying in parallel. Yet practical speedups fall far short of theoretical bounds, especially on reasoning tasks. We identify the root cause as *task isomorphism*: existing methods provide identical prompts to both models, forcing the draft model to solve problems beyond its capacity. This produces high rejection rates that eliminate parallelization gains. We discover that acceptance rates increase monotonically during generation, revealing a latent in-context learning (ICL) effect wherein the draft model progressively aligns with the target distribution by learning from verified tokens. Controlled experiments with varying context window sizes confirm this improvement stems from ICL-driven alignment rather than token difficulty. Motivated by this finding, we propose **Asymmetric Speculative Decoding (ASYMSPEC)**, a training-free framework that decouples the prompts provided to draft and target models. By supplying the draft model with task-relevant example solutions, we reformulate its objective from reasoning to simulation, amplifying the ICL effect from the first token. We prove AsymSpec produces exact samples and converges exponentially to the target distribution. Across reasoning, code generation, and summarization tasks, AsymSpec achieves 2.29× to 4.95× speedup over autoregressive decoding, outperforming standard speculative decoding without model retraining.

1 INTRODUCTION

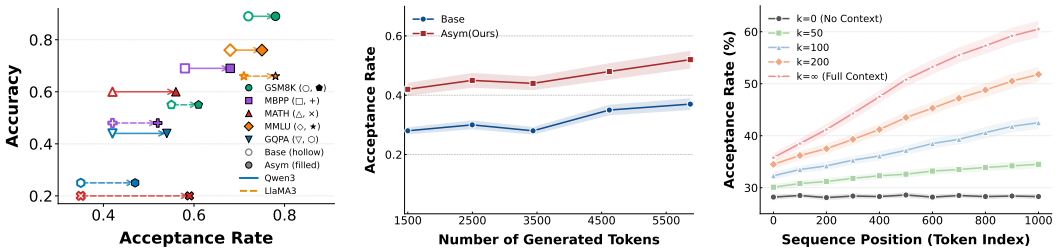
The autoregressive nature of Large Language Models (LLMs) (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023) necessitates sequential token generation, resulting in high inference latency that limits practical deployment (Pope et al., 2023; Kwon et al., 2023). Speculative decoding (Leviathan et al., 2023; Chen et al., 2023) addresses this bottleneck by employing a small draft model to propose candidates and a larger target model verifies in parallel. Although this approach promises substantial speedups, actual performance often falls short of theoretical limits, particularly in reasoning-intensive tasks. (Xia et al., 2024; Sun et al., 2023b; Cai et al., 2024; Li et al., 2024).

We identify the root cause as *task isomorphism*: existing frameworks provide both models with identical prompts, requiring them to solve the same problem independently. When presented with a complex reasoning task (Wei et al., 2022; Cobbe et al., 2021), the draft model's output deviates from the target distribution due to its limited capacity. High rejection rates eliminate the parallelization advantage (Sun et al., 2023b; Yi et al., 2024).

To validate this hypothesis, empirical analysis across reasoning benchmarks and model families reveals that the task isomorphism bottleneck manifests as substantial variation in acceptance rates under symmetric prompting (Figure 1a). Critically, even when draft models demonstrate reasonable standalone task accuracy, acceptance rates remain suboptimal. This gap indicates that the bottleneck lies in distributional alignment rather than reasoning capability. However, asymmetric prompting with behavioral exemplars consistently improves acceptance across diverse tasks, suggesting a mechanism orthogonal to problem-solving capacity. This raises a fundamental question: why does providing exemplars enable distributional alignment?

Investigation of acceptance dynamics throughout the generation process provides critical insight (Figure 1b). In standard speculative decoding, acceptance rates are not uniformly poor but instead improve progressively as generation proceeds. This pattern suggests the draft model learns from

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) **Task isomorphism bottleneck** (b) **Acceptance rates improvement** (c) **Controlling for token difficulty**

Figure 1: **Empirical motivation for asymmetric speculative decoding.** (a) Acceptance rates under Base (standard speculative decoding) and Asym (Ours) across five benchmark tests and two model series. Each point represents a task-model combination. (b) Acceptance rates as a function of generated token in standard speculative decoding (Base) and ASYMSPEC (Ours). (c) Acceptance rates as a function of sequence position under different context window sizes.

accumulated verified history $\mathbf{y}_{<t}$, transitioning from independent problem-solving to trajectory simulation. We interpret this as a latent *in-context learning (ICL) effect* (Brown et al., 2020; Dong et al., 2024; Min et al., 2022). Specifically, the draft model aligns with the target distribution through implicit demonstrations (Dai et al., 2023; Xie et al., 2022) provided by verified tokens.

Controlled experiments confirm that this improvement stems from ICL-driven distributional convergence rather than decreasing token difficulty (Figure 1c). Acceptance rates under varying context window sizes reveal that larger contexts produce steeper improvements across sequence positions. Larger contexts provide more verified history for ICL. With minimal context, acceptance rates remain relatively flat. As context expands, the rate of improvement accelerates markedly. This confirms that ICL-driven alignment, enabled by accumulated verified tokens, underlies the observed phenomenon rather than intrinsic token predictability.

These findings suggest a natural solution: if the draft model can learn to align through accumulated examples, why wait for this alignment to emerge gradually? We propose **Asymmetric Speculative Decoding (ASYMSPEC)**, a framework that provides the ICL effect upfront by decoupling the prompts given to draft and target models. The target model receives the original reasoning task, while the draft model receives an asymmetric prompt enriched with task-relevant example solutions. The framework maintains separate KV caches for each model and employs asymmetric synchronization during verification to preserve exact target distributions.

We establish theoretical correctness by proving that ASYMSPEC produces exact samples through rejection sampling (Theorem 1) and that draft-target distributions converge exponentially as verified history grows (Proposition 2). Across reasoning, code generation, summarization, and instruction following tasks, ASYMSPEC achieves 2.29× to 4.95× speedup over autoregressive decoding on Qwen3 and LLaMA families, consistently outperforming standard speculative decoding and training-free baselines. Compared to training-based method, ASYMSPEC achieves competitive throughput without requiring model retraining. The method demonstrates robustness across various exemplar pools, sampling temperatures and batch sizes. Ablation studies reveal an optimal exemplar lengths that balance ICL effectiveness against prefill overhead.

In summary, our contributions are threefold:

- We identify task isomorphism as the fundamental limitation of speculative decoding on reasoning tasks and reveal a latent in-context learning effect: draft models progressively align with target distributions through accumulated verified context, but this alignment emerges too slowly to benefit early-stage generation.
- We propose ASYMSPEC, a training-free framework that activates the ICL effect from the initial token by providing asymmetric prompts with behavioral exemplars, reformulating the draft model's objective from independent reasoning to trajectory simulation.
- We prove that ASYMSPEC maintains exact correctness through rejection sampling and achieves exponential distributional convergence. Experiments across diverse benchmarks demonstrate 2.29× to 4.95× speedups, matching training-based methods without parameter updates.

2 Related Work

Speculative Decoding. Speculative decoding (Leviathan et al., 2023; Chen et al., 2023) accelerates LLM inference by using a compact draft model to predict tokens verified in parallel by the target model. Recent advances pursue higher acceptance through architectural enhancements (specialized draft models (Yi et al., 2024), multi-head expansions (Cai et al., 2024; Li et al., 2024; 2025), knowledge distillation (Hinton et al., 2015)) or draft-free mechanisms (self-speculation (Zhang et al., 2024), lookahead (Fu et al., 2024), cascaded verification (Kumar et al., 2026)). All existing methods operate under *symmetric prompting* with identical inputs. We identify task isomorphism as the bottleneck and propose asymmetric task reformulation to unlock draft model ICL capacity.

In-Context Learning. ICL enables LLMs to adapt to novel tasks via exemplars without parameter updates (Brown et al., 2020), with theoretical work showing ICL implicitly performs gradient descent (Dai et al., 2023) or Bayesian inference (Xie et al., 2022). While extensively applied to improve standalone model accuracy (Wei et al., 2022; Zhou et al., 2023; Chen et al., 2021), ICL's potential in *collaborative inference* remains unexplored. We leverage ICL to transform draft models from independent solvers into adaptive simulators that predict target model behavior.

Asymmetric Multi-Model Systems. Asymmetric designs have been explored for task-level routing in Mixture-of-Experts (Fedus et al., 2022) and confidence-based cascades (Schuster et al., 2022), but focus on *task-level specialization* rather than *token-level collaboration*. Our framework is the first to systematically exploit prompt asymmetry for speculative decoding, achieving efficiency gains without architectural modifications or training. More related work details are in appendix A.1.

3 Methodology

We formally introduce **Asymmetric Speculative Decoding (AsymSpec)**. We begin by formulating the standard speculative decoding paradigm to highlight its inherent limitations (§3.1). Next, we present empirical evidence revealing a latent in-context learning effect (§3.2), which motivates our framework. We then detail the structural decoupling mechanism (§3.3), establish its theoretical correctness through rigorous analysis (§3.4), and analyze computational complexity (§A.3).

3.1 Preliminaries and Problem Formulation

Standard Speculative Decoding. Let $\mathcal{M}_T$ denote a large target model and $\mathcal{M}_D$ a smaller draft model. Given query $\mathbf{x}$, standard speculative decoding (Leviathan et al., 2023) operates in two phases. First, $\mathcal{M}_D$ generates a candidate sequence $\tilde{Y} = [\tilde{y}_1, \ldots, \tilde{y}_\gamma]$ autoregressively. Second, $\mathcal{M}_T$ evaluates all candidates in parallel via rejection sampling (Leviathan et al., 2023; Chen et al., 2023) , accepting token $\tilde{y}_i$ with probability:

$$\alpha = \min\left(1, \frac{P_T(\tilde{y}_i|\mathbf{x}, \mathbf{y}_{<t}, \tilde{Y}_{<i})}{P_D(\tilde{y}_i|\mathbf{x}, \mathbf{y}_{<t}, \tilde{Y}_{<i})}\right) \tag{1}$$

Speedup depends critically on acceptance rates. High acceptance allows parallel verification to amortize the target model's cost, while low acceptance degrades to sequential generation.

3.2 Motivation: Uncovering the In-Context Learning Bottleneck

Standard speculative decoding enforces a task isomorphism constraint: the draft model must independently solve the same reasoning problem as the target model. We demonstrate that this constraint creates a fundamental bottleneck and reveal a latent in-context learning mechanism that motivates our asymmetric design.

The Task Isomorphism Bottleneck. Acceptance rates across five reasoning benchmarks reveal substantial variation under symmetric prompting (30% to 70%) despite reasonable standalone accuracy (Figure 1a). Critically, even when the draft model achieves 60% accuracy on MATH, acceptance rates remain at 42%. This gap indicates the bottleneck lies in distributional alignment rather than reasoning capability. The draft model reaches correct answers through divergent trajectories that the target model rejects. Asymmetric prompting with behavioral exemplars consistently improves acceptance across all tasks.

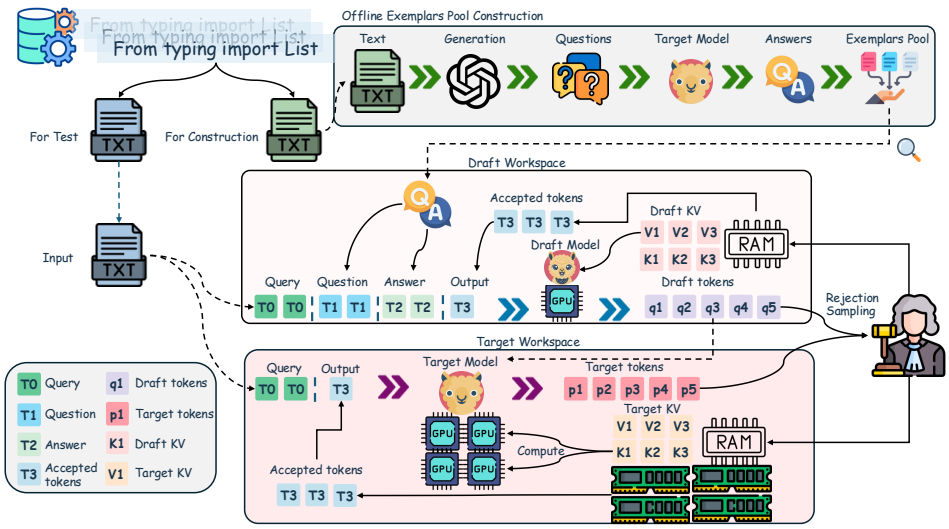


Figure 2: The overall framework of AsymSpec.

Progressive Alignment Through Generation. Acceptance rates rise monotonically from 20% at early tokens to 40% by sequence end under standard speculative decoding (Figure 1b), suggesting the draft model progressively learns from verified history $\mathbf{y}_{<t}$ through in-context learning (ICL). Experiments with varying context window sizes confirm this: larger windows produce steeper acceptance increases, while minimal context yields flat rates (Figure 1c). This disentangles ICL-driven alignment from token difficulty effects.

Design Implication. Standard speculative decoding incurs a cold-start penalty while waiting for ICL to emerge organically. Our solution provides behavioral exemplars upfront, activating the ICL effect from the initial token.

3.3 The AsymSpec Framework

Based on the insight that $\mathcal{M}_D$ excels at trajectory simulation rather than independent reasoning, we introduce AsymSpec, which abandons the task isomorphism constraint by structurally decoupling the contextual states of draft and target models. The overall framework is illustrated in Figure 2.

Offline Exemplars Pool Construction. Since there are no open-source datasets containing demonstrations of the target model's output behavior, we construct an exemplars pool $\mathcal{E}_{\text{pool}}$ offline. Inspired by prompt caching in production APIs, we sample task-relevant questions, generate responses using $\mathcal{M}_T$, and store (c_i, r_i) pairs as behavioral templates. During inference, we retrieve K exemplars from $\mathcal{E}_{\text{pool}}$ based on task alignment to construct $\mathcal{E}_{\text{sim}}$. This one-time construction enables prompt caching when exemplars are reused across queries.

Asymmetric Prompt Formulation. Instead of forcing $\mathcal{M}_D$ to comprehend the complex logic of $\mathbf{x}$, we alter its generation objective through an asymmetric prompt:

$$\mathcal{P}_{\text{draft}} = [\mathcal{I}_{\text{role}} \oplus \mathcal{E}_{\text{sim}} \oplus \mathbf{x}] \tag{2}$$

where $\oplus$ denotes concatenation. This comprises: (1) $\mathcal{I}_{\text{role}}$, a system instruction shifting the draft model's mode from solving to format imitation; (2) $\mathcal{E}_{\text{sim}} = \{(c_1, r_1), \ldots, (c_K, r_K)\}$, K task-aligned behavioral exemplars activating the ICL effect from the first token; and (3) $\mathbf{x}$, the original query. This explicitly conditions the draft distribution $P_D^{\text{sim}}(\cdot \mid \mathcal{P}_{\text{draft}}, \mathbf{y}_{<i})$ on trajectory simulation rather than task resolution.

Decoupled Prefill and Drafting. Because $\mathcal{P}_{\text{draft}} \neq \mathbf{x}$, the framework maintains two physically distinct Key-Value cache prefixes to preserve exact target distributions. During prefill, KV tensors

are computed independently:

$$\mathbf{KV}_T^{(0)} = \text{Prefill}_T(\mathbf{x}) \tag{3}$$

$$\mathbf{KV}_D^{(0)} = \text{Prefill}_D(\mathcal{P}_{\text{draft}}) \tag{4}$$

In the drafting phase, $\mathcal{M}_D$ auto-regressively generates γ candidates:

$$\tilde{y}_i \sim P_D^{\text{sim}}(\tilde{y}_i \mid \mathbf{KV}_D^{(i-1)}), \quad \mathbf{KV}_D^{(i)} = \mathbf{KV}_D^{(i-1)} \cup \text{KV}_D(\tilde{y}_i) \tag{5}$$

Let $\tilde{Y} = [\tilde{y}_1, \ldots, \tilde{y}_\gamma]$ denote the draft sequence.

Parallel Verification and Asymmetric Synchronization. The target model $\mathcal{M}_T$ processes candidates $\tilde{Y}$ in a single parallel forward pass, retrieving from its unmodified task cache $\mathbf{KV}_T$:

$$P_T(\tilde{y}_i \mid \mathbf{x}, \mathbf{y}_{<t}, \tilde{Y}_{<i}) = \text{Forward}_T\left(\tilde{Y} \mid \mathbf{KV}_T^{(t-1)}\right) \tag{6}$$

We employ standard rejection sampling: a draft token $\tilde{y}_i$ is accepted if $r_i \leq \min\left(1, \frac{P_T(\tilde{y}_i|\mathbf{x}, \mathbf{y}_{<t}, \tilde{Y}_{<i})}{P_D^{\text{sim}}(\tilde{y}_i|\mathcal{P}_{\text{draft}}, \mathbf{y}_{<t}, \tilde{Y}_{<i})}\right)$, where $r_i \sim \mathcal{U}(0,1)$.

When rejection occurs at the k-th token, the framework accepts prefix $\tilde{Y}_{<k}$ and $\mathcal{M}_T$ samples a corrective token y_{corr} from its residual distribution. The target cache advances by absorbing the verified sequence:

$$\mathbf{KV}_T^{(t+k)} = \mathbf{KV}_T^{(t-1)} \cup \text{KV}_T(\tilde{Y}_{<k} \oplus y_{\text{corr}}) \tag{7}$$

The draft cache performs asymmetric rollback, truncating the invalid trajectory $\tilde{Y}_{k:\gamma}$ and synchronizing with y_{corr}:

$$\mathbf{KV}_D^{(t+k)} = \text{Truncate}(\mathbf{KV}_D^{(t+\gamma)}, k-1) \cup \text{KV}_D(y_{\text{corr}}) \tag{8}$$

This guarantees that both models enter the next iteration with identical historical text sequences.

3.4 Theoretical Analysis

Correctness. Despite using different prompts, AsymSpec produces exact samples from the target distribution.

Theorem 1 (Correctness of AsymSpec)**.** *For any verified history $\mathbf{y}_{<t}$,* AsymSpec *produces exact samples from the target distribution:*

$$P_{\textit{AsymSpec}}(y_t \mid \mathbf{x}, \mathbf{y}_{<t}) = P_T(y_t \mid \mathbf{x}, \mathbf{y}_{<t}) \tag{9}$$

Proof. The correctness follows from rejection sampling Chen et al. (2023); Leviathan et al. (2023). We accept draft token $\tilde{y}_i$ with probability:

$$\alpha = \min\left(1, \frac{P_T(\tilde{y}_i \mid \mathbf{x}, \mathbf{y}_{<t}, \tilde{Y}_{<i})}{P_D^{\text{sim}}(\tilde{y}_i \mid \mathcal{P}_{\text{draft}}, \mathbf{y}_{<t}, \tilde{Y}_{<i})}\right) \tag{10}$$

By the fundamental theorem of rejection sampling (Kim, 2000), the output distribution equals the target distribution exactly. While the draft model uses asymmetric prompt $\mathcal{P}_{\text{draft}}$ for generation, the target model always evaluates candidates using the original prompt $\mathbf{x}$, maintaining exact correctness. □

Effectiveness. Higher acceptance rates arise from distribution convergence as verified history $\mathbf{y}_{<t}$ grows. Two Transformer properties drive this:

(i) **Attention Decay:** Attention on initial prompt tokens decays exponentially (Press et al., 2022; Sun et al., 2023a):

$$\sum_{i=1}^{|\mathbf{x}|} \text{Attn}(q_t, k_i) \leq \beta \cdot e^{-\lambda t} \tag{11}$$

(ii) **Lipschitz Continuity:** Output distributions are Lipschitz continuous (Gao et al., 2018):

$$\text{TV}(P(\cdot \mid h), P(\cdot \mid h')) \leq L\|h - h'\|_2 \tag{12}$$

Table 1: Performance comparison across reasoning and coding benchmarks. Speedup (S) and acceptance length (τ) are reported relative to Vanilla AR baseline (1.00×). ASYMSPEC consistently outperforms Standard Speculative Decoding (SPS) and Speculative-Speculative Decoding (SSD) across all model pairs and tasks, with gains scaling favorably on larger capacity gaps. **Bold** indicates best performance in each column.

Model	Method	MATH-500		AIME25		HumanEval		CNN/DM		MT-Bench		Average	
		S	τ	S	τ	S	τ	S	τ	S	τ	S	τ
	Vanilla	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Q3-8B	SPS	1.81	4.87	1.75	4.75	1.80	4.82	1.67	4.68	1.71	4.58	1.74	4.74
	SSD	2.16	5.26	2.08	5.08	2.09	5.19	1.96	5.11	1.93	4.90	2.04	5.10
	Ours	**2.45**	**6.13**	**2.14**	**5.40**	**2.42**	**6.10**	**2.41**	**6.35**	**2.06**	**5.38**	**2.29**	**5.87**
Q3-32B	SPS	1.94	3.55	2.54	4.65	2.24	4.12	1.81	3.42	1.94	3.55	2.09	3.85
	SSD	2.71	3.68	4.08	5.02	4.22	5.20	2.72	3.71	2.74	3.68	3.29	4.25
	Ours	**3.91**	**5.22**	**4.32**	**5.51**	**4.89**	**6.11**	**4.21**	**5.78**	**4.03**	**5.22**	**4.27**	**5.57**
L3.1-8B	SPS	1.63	5.49	1.50	5.10	1.69	5.69	1.32	4.51	1.46	4.90	1.52	5.13
	SSD	1.88	6.03	1.70	5.55	1.96	6.26	1.87	6.21	1.60	5.30	1.80	5.87
	Ours	**2.05**	**6.63**	**1.80**	**5.92**	**2.05**	**6.55**	**2.17**	**7.02**	**2.67**	**6.31**	**2.14**	**6.48**
L3.1-70B	SPS	3.40	5.63	2.99	5.04	3.26	5.61	2.66	3.69	2.69	4.62	3.00	4.98
	SSD	5.52	6.19	4.42	5.47	4.99	6.22	2.93	4.09	3.82	4.92	4.33	5.38
	Ours	**5.65**	**6.41**	**4.53**	**5.61**	**5.35**	**6.68**	**4.74**	**6.60**	**4.52**	**5.95**	**4.95**	**6.25**

Proposition 2 (Distribution Convergence)**.** *As verified history length t increases, draft and target distributions converge exponentially:*

$$TV\left(P_D(\cdot \mid \mathcal{P}_{draft}, \mathbf{y}_{<t}), P_D(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})\right) \leq C \cdot e^{-\lambda t} \tag{13}$$

This explains effectiveness: early tokens benefit from behavioral exemplars $\mathcal{E}_{\text{sim}}$ activating ICL; later tokens benefit from attention decay making both models focus on recent history $\mathbf{y}_{<t}$ rather than initial prompts. Please see the appendix A.2 for more details.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We evaluate ASYMSPEC on state-of-the-art open-weight LLMs. Target-draft pairs include Qwen-3 (8B/32B), and LLaMA-3.1 (8B/70B).

Datasets. We select benchmarks spanning mathematical reasoning, code generation, and instruction following: (1) *Mathematics*: AIME (Lightman et al., 2024), MATH-500 (Hendrycks et al., 2021); (2) *Code*: HumanEval (Chen et al., 2021); (3) *Summarization*: CNN/DM (See et al., 2017); (4) *Instruction*: MT-Bench (Zheng et al., 2023).

Baselines. Our method belongs to the family of training-free methods, which uses a small draft model from the same series. Therefore, we compare our method with Vanilla Autoregressive Decoding and Standard Speculative Decoding (SPS) (Leviathan et al., 2023; Chen et al., 2023)). We also compare it with the state-of-the-art training-free SOTA method, Speculative Decoding (SSD) (Kumar et al., 2026). We also introduce Eagle-3 (Li et al., 2025) as a training-required baseline.

Metrics and Implementation. We measure wall-clock speedup (S) relative to vanilla autoregressive decoding and mean acceptance length (τ) per speculative iteration. Experiments were conducted on NVIDIA RTX 4090 and A6000 GPUs. We integrated ASYMSPEC into `nano-vllm` and used *PageAttention* (Kwon et al., 2023) to manage the key-value cache.

4.2 MAIN RESULTS

Table 1 presents speedup and acceptance length across all benchmarks and model pairs. ASYMSPEC consistently outperforms both Standard Speculative Decoding (SPS) and Speculative-Speculative

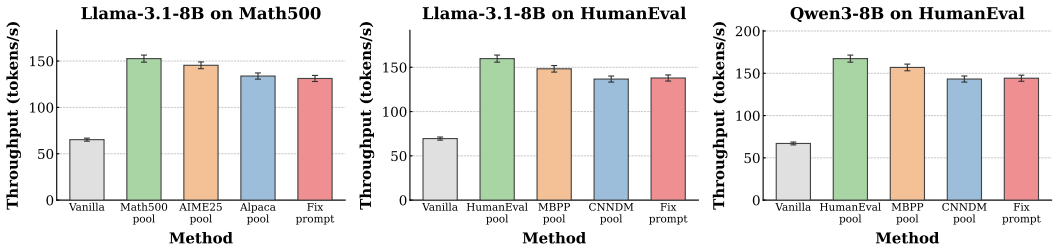


Figure 3: Throughput comparison across different draft pool construction strategies on LLaMA-3.1-8B (MATH-500 and HumanEval) and Qwen-3-8B (HumanEval) at $B = 1, T = 0.0$. Task-specific draft pools consistently achieve the highest throughput, with HumanEval-derived and Math500-derived pools outperforming generic alternatives by 5–8%.

Decoding (SSD) across all configurations, with performance gains scaling with model capacity. On small model pairs (Qwen-3-8B), ASYMSPEC achieves 2.29× average speedup versus 1.74× for SPS, while on large model pairs (LLaMA-3.1-70B), gains increase to 4.95× versus 3.00× for SPS. This scaling behavior reflects the core mechanism: larger capacity gaps amplify the benefits (Hinton et al., 2015; Gou et al., 2021) of asymmetric prompting, as draft models leverage behavioral exemplars to compensate for reasoning limitations.

ASYMSPEC substantially improves task consistency compared to symmetric baselines. On Qwen-3-8B, SPS exhibits high variance across tasks (1.67×–1.81×), while ASYMSPEC narrows this range to 2.06×–2.45×, demonstrating that asymmetric prompting mitigates the task isomorphism trap. Compared to SSD, which achieves a 4.33x speedup on the LLaMA-3.1-70B dataset, ASYMSPEC delivers a 4.95× speedup, representing a 0.62× improvement in speedup ratio.

4.3 SYSTEM ROBUSTNESS

Impact of Draft Pool Construction. Figure 3 demonstrates that task-specific draft pools consistently outperform generic alternatives across all configurations on LLaMA-3.1-8B (MATH-500, HumanEval) and Qwen-3-8B (HumanEval). On HumanEval (LLaMA-3.1-8B), the HumanEval-derived pool achieves 159.8 tokens/s versus 148.3 tokens/s for MBPP pool (7.8% improvement) and 136.7 tokens/s for CNNDM pool. On MATH-500, the Math500-derived pool reaches 152.6 tokens/s compared to 145.4 tokens/s for AIME25 pool (4.9% improvement) and 133.8 tokens/s for Alpaca pool. Similar patterns hold on Qwen-3-8B HumanEval (167.4 tokens/s vs. 156.9 tokens/s for MBPP, 6.7% improvement). Notably, even the fixed prompt baseline (137.9–144.2 tokens/s) outperforms vanilla decoding (65.2–69.5 tokens/s) by 2.0–2.2×, confirming that ASYMSPEC's core mechanism benefits from any reasonable draft source. The critical advantage lies in capturing domain-specific patterns: code benchmarks benefit from code-derived drafts, while math tasks gain from math-oriented examples, enabling dynamic adaptation to diverse workloads without retraining.

Temperature Robustness. Figure 4 demonstrates performance across $T \in [0.0, 1.0]$ (Holtzman et al., 2020) on LLaMA-3.1-8B (MATH-500, HumanEval) and Qwen-3-8B (HumanEval). On MATH-500, Standard Speculative Decoding (SPS) degrades from 1.63× to 1.23× (24.2% drop) and Asynchronous Speculative Decoding (SSD) from 1.87× to 1.28× (31.7% drop), while ASYMSPEC maintains 2.04× at $T = 0.0$ and 1.35× at $T = 1.0$, preserving substantial advantage over baselines. On HumanEval (LLaMA-3.1-8B), ASYMSPEC declines from 2.00× to 1.40× versus SPS (1.69× to 1.17×) and SSD (1.96× to 1.25×). The consistent performance gap validates that behavioral exemplars anchor draft predictions to target patterns, mitigating distribution shift under stochastic sampling.

Batch Size Scalability. Figure 5 demonstrates that ASYMSPEC maintains throughput advantages across batch sizes ($B \in \{1, 2, 4, 8, 16\}$) (Yu et al., 2022) on LLaMA-3.1-8B (MATH-500, HumanEval) and Qwen-3-8B (HumanEval). On MATH-500, ASYMSPEC achieves 163.2 tokens/s at $B = 1$ versus 120.9 tokens/s for SPS (35.0% improvement), scaling to 1281.3 tokens/s at $B = 16$ versus 1107.2 tokens/s for SPS (15.7% advantage). Similar trends hold across configurations: on HumanEval (LLaMA-3.1-8B), ASYMSPEC reaches 1241.3 tokens/s at $B = 16$ (11.2% over SPS). The narrowing relative advantage at larger batch sizes reflects GPU saturation, where the bottleneck shifts from computation to memory bandwidth. However, ASYMSPEC maintains absolute through-

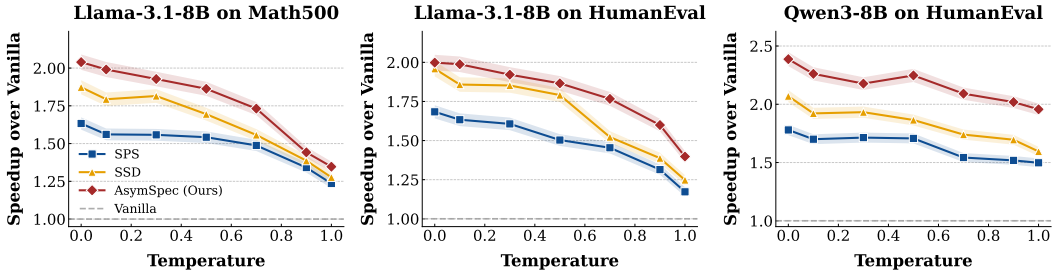


Figure 4: Speedup ratio as a function of sampling temperature. We compare SPS, SSD, and AsymSpec across $T \in [0.0, 1.0]$ on LLaMA-3.1-8B with MATH-500 (left) and HumanEval (middle), and Qwen-3-8B on HumanEval (right). All methods show performance degradation as temperature increases, with AsymSpec maintaining the highest speedup across all settings.

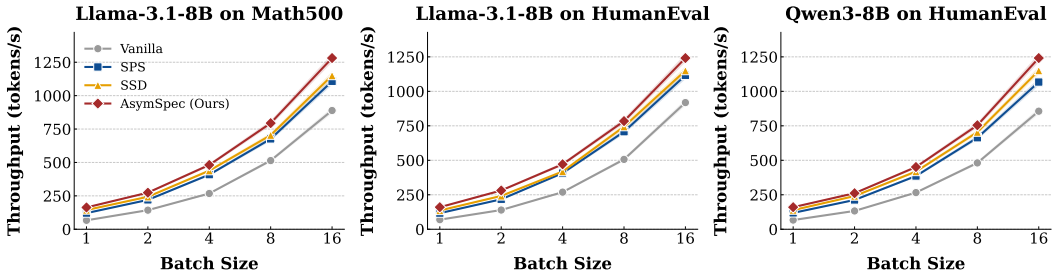


Figure 5: Throughput (tokens/s) as a function of batch size on LLaMA-3.1-8B (MATH-500 and HumanEval) and Qwen-3-8B (HumanEval). AsymSpec consistently outperformed the baseline across all batch sizes, demonstrating its robustness under various deployment constraints.

put leadership across all batch sizes, with prefix caching (Zheng et al., 2024; Kwon et al., 2023) amortizing exemplar processing overhead in production deployments. The consistent throughput gains validate that asymmetric prompting scales effectively in production batched inference scenarios. The relative speedup advantage narrows at larger batch sizes (from 35.0% at $B = 1$ to 15.7% at $B = 16$), reflecting GPU saturation where memory bandwidth becomes the bottleneck (Pope et al., 2023; Shazeer, 2019). However, AsymSpec maintains absolute throughput leadership, confirming robustness under varying deployment constraints.

4.4 Comparison with State-of-the-Art

We compare AsymSpec against training-based EAGLE-3 and asynchronous SSD on identical hardware (NVIDIA A6000, $B = 1$, $T = 0.0$). Figure 6 shows that AsymSpec achieves competitive or superior throughput across all configurations. On LLaMA-3.1-8B with MATH-500, AsymSpec reaches 163.2 tokens/s versus 125.2 tokens/s for EAGLE-3 (30.3% improvement) and 143.8 tokens/s for SSD. Similar trends hold on HumanEval: AsymSpec achieves 159.9 tokens/s versus 128.5 tokens/s for EAGLE-3 (24.4% improvement). These results demonstrate that ICL-based alignment rivals or exceeds specialized draft heads trained on target outputs, while maintaining the advantage of being training-free. The critical distinction lies in deployment costs: training-based methods require substantial upfront investment (EAGLE-3 (Li et al., 2025) needs 100 GPU-hours on 8× A100) and must retrain for new domains. In contrast, AsymSpec requires only representative samples per task, enabling seamless integration and dynamic adaptation without modifying model weights.

4.5 Ablation Study: Impact of Extra Prompt Length

We quantify the tradeoff between ICL effectiveness and prefill overhead by measuring throughput, acceptance length, and acceptance rate across extra prompt lengths from 32 to 2048 tokens. Figure 7 presents results on Qwen-3-8B and LLaMA-3.1-8B (HumanEval and MATH-500). All configurations exhibit a consistent inverted-U relationship across all three metrics. On Qwen-3-8B with MATH-

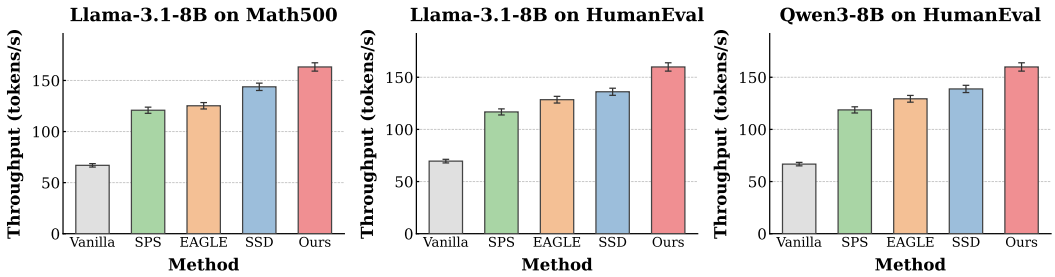


Figure 6: Throughput comparison with state-of-the-art methods on LLaMA-3.1-8B (MATH-500 and HumanEval) and Qwen-3-8B (HumanEval) at $B = 1$, $T = 0.0$. AsymSpec outperforms training-based EAGLE-3 and asynchronous SSD across all configurations.

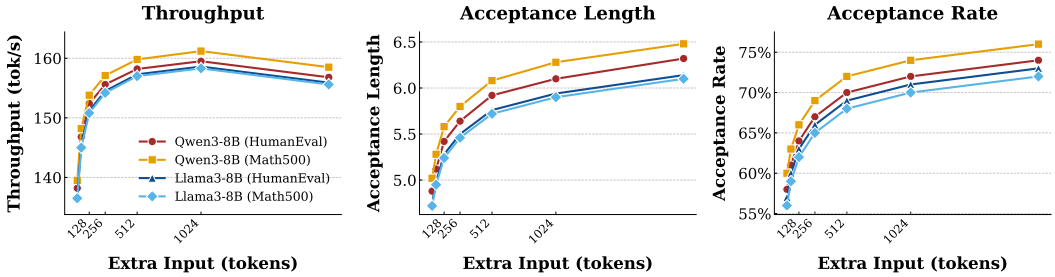


Figure 7: Impact of extra input length on AsymSpec performance. Left: throughput (tokens/s), Middle: acceptance length (tokens), Right: acceptance rate (%). We evaluate Qwen-3-8B and LLaMA-3.1-8B on HumanEval and MATH-500 benchmarks.

500, throughput increases from 139.5 tokens/s at 32 tokens to a peak of 161.2 tokens/s at 1024 tokens (15.5% improvement), then declines to 158.5 tokens/s at 2048 tokens. Acceptance length grows monotonically from 5.02 to 6.48 tokens (29.1% improvement), and acceptance rate increases from 60% to 76%. Critically, acceptance metrics continue improving beyond 1024 tokens while throughput degrades, demonstrating that ICL effectiveness does not saturate but prefill latency eventually dominates. The optimal prompt length of 512–1024 tokens provides actionable deployment guidance: tune prompt length based on task complexity and hardware constraints, targeting the regime where ICL effectiveness peaks before prefill overhead dominates.

5 Conclusion and Future Work

We have introduced AsymSpec, a training-free framework that breaks the symmetric input constraint in speculative decoding by providing draft and target models with structurally different prompts. By identifying task isomorphism as a fundamental bottleneck and leveraging the in-context learning effect in draft models, AsymSpec reframes the draft model's objective from independent problem-solving to trajectory simulation. This design achieves up to 4.95× speedup over autoregressive decoding on reasoning benchmarks, consistently outperforming standard speculative decoding across multiple model families without requiring retraining or architectural modifications.

The asymmetric design introduces additional input to the draft model and necessitates maintaining two separate KV caches, effectively doubling the memory footprint relative to standard speculative decoding. When auxiliary input grows excessively large, the expanded KV cache can introduce latency that counteracts the benefits of higher acceptance rates, particularly under memory-bandwidth constraints. This trade-off requires careful calibration of auxiliary prompt size against available hardware resources. Prefix caching can mitigate this overhead by amortizing the cost of common exemplars across multiple requests, making the approach viable for production systems. Future directions include adaptive exemplar selection strategies that dynamically adjust input length in response to real-time system conditions, and shared KV cache architectures that consolidate draft and target caches to integrate seamlessly with existing inference infrastructure.

AI USE STATEMENT

We used generative AI tools for text revision, experimental code assistance, and visualization code generation, but all core contributions, experimental design, and final manuscript were produced by the authors, who take full responsibility for the work.

ETHICS STATEMENT

This work presents a training-free inference optimization technique that does not involve human subjects or introduce ethical concerns beyond those of existing LLM deployment, and provides no additional capability for harmful content generation beyond what the underlying models already possess.

REPRODUCIBILITY STATEMENT

All implementation details, model configurations, and experimental settings are provided in the paper, and we will release complete source code upon publication to ensure full reproducibility with publicly available models and datasets.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021. URL https://arxiv.org/abs/2108.07732.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 5209–5235. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/cai24b.html.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318, 2023. doi: 10.48550/ARXIV.2302.01318. URL https://doi.org/10.48550/arXiv.2302.01318.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob

McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

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. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, Findings of ACL, pp. 4005–4019. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.247. URL https://doi.org/10.18653/v1/2023.findings-acl.247.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 1107–1128. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.64. URL https://doi.org/10.18653/v1/2024.emnlp-main.64.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of LLM inference using lookahead decoding. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 14060–14079. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/fu24a.html.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE Trans. Inf. Theory*, 64(8):5629–5661, 2018. doi: 10.1109/TIT.2018.2807481. URL https://doi.org/10.1109/TIT.2018.2807481.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *Int. J. Comput. Vis.*, 129(6):1789–1819, 2021. doi: 10.1007/S11263-021-01453-Z. URL https://doi.org/10.1007/s11263-021-01453-z.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Hoon Kim. Monte carlo statistical methods. *Technometrics*, 42(4):430–431, 2000. doi: 10.1080/00401706.2000.10485722. URL https://doi.org/10.1080/00401706.2000.10485722.

Tanishq Kumar, Tri Dao, and Avner May. Speculative speculative decoding. *CoRR*, abs/2603.03251, 2026. doi: 10.48550/ARXIV.2603.03251. URL https://doi.org/10.48550/arXiv.2603.03251.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (eds.), *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: speculative sampling requires rethinking feature uncertainty. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 28935–28948. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/li24bt.html.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: scaling up inference acceleration of large language models via training-time test. *CoRR*, abs/2503.01840, 2025. doi: 10.48550/ARXIV.2503.01840. URL https://doi.org/10.48550/arXiv.2503.01840.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 11048–11064. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.759. URL https://doi.org/10.18653/v1/2022.emnlp-main.759.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In Dawn Song, Michael Carbin, and Tianqi Chen (eds.), *Proceedings of the Sixth Conference on Machine Learning and Systems, MLSys 2023, Miami, FL, USA, June 4-8, 2023*. mlsys.org, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/c4be71ab8d24cdfb45e3d06dbfca2780-Abstract-mlsys2023.html.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6fac9e316a4ae75ea244ddcef1982c71-Abstract-Conference.html.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1073–1083. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1099. URL https://doi.org/10.18653/v1/P17-1099.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *CoRR*, abs/1911.02150, 2019. URL http://arxiv.org/abs/1911.02150.

Benjamin Spector and Christopher Ré. Accelerating LLM inference with staged speculative decoding. *CoRR*, abs/2308.04623, 2023. doi: 10.48550/ARXIV.2308.04623. URL https://doi.org/10.48550/arXiv.2308.04623.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 14590–14604. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.ACL-LONG.816. URL https://doi.org/10.18653/v1/2023.acl-long.816.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix X. Yu. Spectr: Fast speculative decoding via optimal transport. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/6034a661584af6c28fd97a6f23e56c0a-Abstract-Conference.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pp. 7655–7671. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.456. URL https://doi.org/10.18653/v1/2024.findings-acl.456.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Euiin Yi, Taehyeon Kim, Hongseok Jeung, Du-Seong Chang, and Se-Young Yun. Towards fast multilingual LLM inference: Speculative decoding and specialized drafters. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 10789–10802. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.602. URL https://doi.org/10.18653/v1/2024.emnlp-main.602.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for transformer-based generative models. In Marcos K. Aguilera and Hakim Weatherspoon (eds.), *16th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2022, Carlsbad, CA, USA, July 11-13, 2022*, pp. 521–538. USENIX Association, 2022. URL https://www.usenix.org/conference/osdi22/presentation/yu.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft& verify: Lossless large language model acceleration via self-speculative decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 11263–11282. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.607. URL https://doi.org/10.18653/v1/2024.acl-long.607.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W. Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/724be4472168f31ba1c9ac630f15dec8-Abstract-Conference.html.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

A Appendix

A.1 Details of related work

A.1.1 Speculative Decoding for LLM Inference

Speculative decoding (Leviathan et al., 2023; Chen et al., 2023) has emerged as a promising paradigm for accelerating autoregressive LLM inference by leveraging a compact draft model to predict future tokens that are verified in parallel by a larger target model. This approach preserves the exact output distribution of the target model while reducing wall-clock latency through parallel verification (Xia et al., 2024). Recent advances pursue higher acceptance rates via two primary directions: (1) **Architectural Enhancement**, which trains specialized draft models (Yi et al., 2024), employs multi-head expansions such as Medusa (Cai et al., 2024) and EAGLE (Li et al., 2024; 2025), or applies knowledge distillation (Hinton et al., 2015; Gou et al., 2021) to align draft models with target distributions; and (2) **Draft-free Mechanisms**, which exploit self-speculation (Zhang et al., 2024), lookahead heuristics (Fu et al., 2024). Staged speculative decoding (Spector & Ré, 2023) and cascaded verification (Kumar et al., 2026) further improve efficiency through hierarchical draft generation. However, all existing methods operate under *symmetric prompting*, where draft and target models receive identical inputs. We identify task isomorphism as the fundamental bottleneck limiting draft model effectiveness and propose asymmetric task reformulation as a training-free alternative that unlocks the in-context learning capacity of draft models.

A.1.2 In-Context Learning and Few-Shot Adaptation

In-context learning (ICL) enables LLMs to perform novel tasks by conditioning on a few exemplars without parameter updates (Brown et al., 2020). Theoretical investigations reveal that ICL implicitly performs gradient descent (Dai et al., 2023) or Bayesian inference (Xie et al., 2022), allowing models to adapt to new distributions at inference time. Empirical studies demonstrate that ICL effectiveness depends on exemplar quality (Min et al., 2022), format consistency, and the number of demonstrations (Dong et al., 2024). While ICL has been extensively applied to improve standalone model accuracy on reasoning (Wei et al., 2022; Zhou et al., 2023), code generation (Chen et al., 2021; Austin et al., 2021), and mathematical problem-solving (Cobbe et al., 2021; Lightman et al., 2024), its potential in *collaborative inference* remains unexplored. We leverage the emergent ICL effect in draft models to align their output distribution with the target model. This transforms draft models from independent solvers into adaptive simulators that predict target model behavior rather than ground-truth solutions.

A.1.3 Prompt Engineering and Asymmetric Multi-Model Systems

Prompt engineering techniques such as Chain-of-Thought (Wei et al., 2022), Least-to-Most prompting (Zhou et al., 2023), and step-by-step verification (Lightman et al., 2024) have been developed to maximize reasoning depth and accuracy in single-agent systems. In multi-model architectures, asymmetric designs have been explored for task-level routing in Mixture-of-Experts (Fedus et al., 2022), confidence-based cascades (Schuster et al., 2022), and distributed serving systems (Yu et al., 2022). However, these approaches focus on *task-level specialization* rather than *token-level collaboration* where models jointly decode the same sequence. Our framework is the first to systematically exploit *prompt asymmetry* for speculative decoding, intentionally decoupling the instruction space of draft and target models to optimize their respective roles. By providing the draft model with target-generated exemplars while the target model solves the original task, we demonstrate that asymmetric prompting unlocks substantial efficiency gains without architectural modifications or additional training, distinguishing our approach from prior work on model distillation (Hinton et al., 2015) or fine-tuning-based alignment (Yi et al., 2024).

A.2 Complete Proof of Proposition 2

We provide the complete proof of Proposition 2, which establishes the exponential convergence of the draft model's distribution under asymmetric prompting. We proceed through a series of lemmas that decompose the problem into manageable components.

A.2.1 LEMMA 1: PROMPT INFLUENCE DECAY

Lemma 3 (Prompt Influence Decay)**.** *For a Transformer model with L layers and attention mechanism, the influence of the initial prompt $\mathbf{x}$ on the hidden state at position t decays exponentially:*

$$\|h_t(\mathbf{x}, \mathbf{y}_{<t}) - h_t(\mathbf{x}', \mathbf{y}_{<t})\|_2 \leq \beta \cdot e^{-\lambda t} \cdot \|\mathbf{x} - \mathbf{x}'\|_2 \tag{14}$$

for any two prompts $\mathbf{x}, \mathbf{x}'$ and verified history $\mathbf{y}_{<t}$.

Proof. Consider the attention mechanism at layer ℓ and position t. The hidden state is computed as:

$$h_t^{(\ell)} = \text{Attn}(h_t^{(\ell-1)}, [h_{\mathbf{x}}^{(\ell-1)}, h_{\mathbf{y}_{<t}}^{(\ell-1)}]) + \text{FFN}(h_t^{(\ell-1)}) \tag{15}$$

The attention weights on the prompt tokens are:

$$\alpha_{t,i}^{(\ell)} = \frac{\exp(q_t^{(\ell)\top} k_i^{(\ell)} / \sqrt{d})}{\sum_j \exp(q_t^{(\ell)\top} k_j^{(\ell)} / \sqrt{d})} \tag{16}$$

For causal attention with positional encodings, empirical studies (Sun et al., 2023a; Press et al., 2022) show that attention weights decay approximately exponentially with distance:

$$\alpha_{t,i} \approx \frac{1}{Z} \exp\left(-\frac{t-i}{\tau}\right) \tag{17}$$

where τ is the attention decay constant and Z is a normalization factor.

Therefore, the contribution from prompt tokens is:

$$\left\| \sum_{i \in \mathbf{x}} \alpha_{t,i} v_i \right\|_2 \leq \sum_{i \in \mathbf{x}} \alpha_{t,i} \|v_i\|_2 \tag{18}$$

$$\leq \|v\|_{\max} \sum_{i \in \mathbf{x}} \frac{1}{Z} \exp\left(-\frac{t-i}{\tau}\right) \tag{19}$$

$$\leq \frac{\|v\|_{\max}}{Z} \cdot \frac{|\mathbf{x}|}{1 - e^{-1/\tau}} \cdot e^{-t/\tau} \tag{20}$$

Setting $\beta = \frac{\|v\|_{\max}|\mathbf{x}|}{Z(1-e^{-1/\tau})}$ and $\lambda = 1/\tau$, we obtain the desired exponential decay. Propagating through L layers introduces a multiplicative factor, but the exponential decay is preserved. □

Remark. The exponential attention decay in Equation (34) is an empirical approximation that holds well in practice for causal Transformers with positional encodings, as demonstrated by Sun et al. (2023a); Press et al. (2022). While not a strict mathematical guarantee, this property has been consistently observed across diverse model architectures and provides a principled explanation for the convergence behavior.

A.2.2 LEMMA 2: DISTRIBUTION CONTINUITY

Lemma 4 (Distribution Continuity)**.** *The output distribution of a Transformer language model is Lipschitz continuous with respect to the final hidden state:*

$$TV(P(\cdot \mid h), P(\cdot \mid h')) \leq L\|h - h'\|_2 \tag{21}$$

where $L = \frac{\|W_{out}\|_2}{\sqrt{2}}$ and W_{out} is the output projection matrix.

Proof. The output distribution is computed via softmax:

$$P(y \mid h) = \frac{\exp(w_y^\top h)}{\sum_{y'} \exp(w_{y'}^\top h)} \tag{22}$$

The total variation distance between two categorical distributions is:

$$\text{TV}(P, P') = \frac{1}{2} \sum_{y} |P(y) - P'(y)| \tag{23}$$

Using the fact that softmax is 1-Lipschitz with respect to the ℓ_1 norm of logits, and the logit difference is bounded by:

$$|w_y^\top h - w_y^\top h'| \leq \|w_y\|_2 \|h - h'\|_2 \leq \|W_{\text{out}}\|_2 \|h - h'\|_2 \tag{24}$$

Applying Pinsker's inequality and standard concentration results for softmax (Gao et al., 2018), we obtain:

$$\text{TV}(P(\cdot \mid h), P(\cdot \mid h')) \leq \frac{\|W_{\text{out}}\|_2}{\sqrt{2}} \|h - h'\|_2 \tag{25}$$

□

A.2.3 Proof of Proposition 2

We now combine the lemmas to prove Proposition 2.

Proof of Proposition 2. We bound the total variation distance between the draft model's distributions under two different prompts: $\mathcal{P}_{\text{draft}}$ (asymmetric prompt) and $\mathbf{x}$ (original prompt), both conditioned on the same verified history $\mathbf{y}_{<t}$.

Step 1: Bound the hidden state difference. By Lemma 3, the hidden states of the draft model under the two prompts satisfy:

$$\|h_t^D(\mathcal{P}_{\text{draft}}, \mathbf{y}_{<t}) - h_t^D(\mathbf{x}, \mathbf{y}_{<t})\|_2 \leq \beta \cdot e^{-\lambda t} \cdot \|\mathcal{P}_{\text{draft}} - \mathbf{x}\|_2 \tag{26}$$

Since $\mathcal{P}_{\text{draft}} = [\mathcal{I}_{\text{role}} \oplus \mathcal{E}_{\text{sim}} \oplus \mathbf{x}]$, the prompt difference is bounded by the length of the added prefix:

$$\|\mathcal{P}_{\text{draft}} - \mathbf{x}\|_2 \leq \sqrt{|\mathcal{I}_{\text{role}}| + |\mathcal{E}_{\text{sim}}|} \cdot \|e\|_2 \leq C_{\text{prompt}} \tag{27}$$

where $\|e\|_2$ is the norm of token embeddings and C_{prompt} is a constant.

Step 2: Bound the distribution difference. By Lemma 4, the distributional difference is:

$$\begin{aligned} &\text{TV}(P_D(\cdot \mid \mathcal{P}_{\text{draft}}, \mathbf{y}_{<t}), P_D(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})) && (28)\\ &\leq L \cdot \|h_t^D(\mathcal{P}_{\text{draft}}, \mathbf{y}_{<t}) - h_t^D(\mathbf{x}, \mathbf{y}_{<t})\|_2 && (29)\\ &\leq L\beta C_{\text{prompt}} \cdot e^{-\lambda t} && (30) \end{aligned}$$

Setting $C = L\beta C_{\text{prompt}}$ completes the proof. □

Remark on Correctness. Note that Proposition 2 concerns the convergence of draft model distributions, not the correctness of AsymSpec's output. The correctness of AsymSpec (Theorem 1) follows directly from rejection sampling theory and does not depend on this convergence result. The convergence result explains *why* AsymSpec achieves high acceptance rates, particularly for long sequences.

A.3 Complexity and Overhead Analysis

Latency Decomposition. Compared to traditional speculative decoding, AsymSpec introduces additional costs but yields greater savings:

$$\Delta T_{\text{total}} = L_{\text{ex}} \cdot M_D + \gamma \cdot M_D \cdot (L_{\text{ex}}/L_{\text{ctx}}) - N \cdot M_T \cdot (\alpha_{\text{asym}} - \alpha_{\text{base}})/\gamma \tag{31}$$

where L_{ex} is exemplar length, γ is draft length, L_{ctx} is context length, N is output length, and $\alpha_{\text{asym}}, \alpha_{\text{base}}$ are acceptance rates of AsymSpec and traditional speculative decoding. The first two terms represent prefill and draft decoding overhead; the third term is time saved from reduced target model iterations. The draft overhead becomes negligible as context grows.

Speedup Condition. Net speedup occurs when $\Delta T_{\text{total}} < 0$. This favors long-sequence generation where savings scale linearly with N while costs remain constant. Prefix caching (Zheng et al., 2024) amortizes prefill overhead to near zero across requests.

Memory Overhead. Asymmetric prompts add KV cache:

$$\Delta_{\text{mem}} = 2 \times L_{\text{ex}} \times d \times n_{\text{layers}} \times \text{sizeof(dtype)} \tag{32}$$

This is constant per request. While prefix caching (Zheng et al., 2024) allows sharing across requests, the memory overhead remains but the computational cost is negligible for typical configurations.

A.4 Algorithm Implementations

The entire AsymSpec pipeline is training-free and conceptually elegant. Algorithm 1 presents the complete implementation, with modifications from standard speculative decoding highlighted in blue. The key changes are: (1) asymmetric prompt construction, (2) decoupled KV cache maintenance, and (3) asymmetric cache synchronization after verification. For comparison, the standard baseline is provided in Algorithm 2.

Algorithm 1 AsymSpec decoding algorithm. Modifications from standard speculative decoding are highlighted in blue, including asymmetric prompt construction (line 1), decoupled KV cache management (lines 2, 7), and asymmetric cache synchronization after verification (line 15). The framework maintains exact target distribution while enabling ICL-driven draft alignment.

Require: Target $\mathcal{M}_T$, Draft $\mathcal{M}_D$, Original prompt $\mathbf{x}$, Draft budget γ

1: $\mathcal{P}_{\text{draft}} \leftarrow \text{ConstructAsymPrompt}(\mathbf{x})$
2: $\mathbf{KV}_T \leftarrow \text{Prefill}_T(\mathbf{x})$, $\mathbf{KV}_D \leftarrow \text{Prefill}_D(\mathcal{P}_{\text{draft}})$
3: $\mathbf{y} \leftarrow [\,]$
4: **while** generation not finished **do**
5: $\quad\tilde{Y} \leftarrow [\,], k \leftarrow 0$
6: $\quad$**for** $i = 1$ to γ **do**
7: $\quad\quad$Sample $\tilde{y}_i \sim P_D^{\text{sim}}(\cdot \mid \mathbf{KV}_D), \mathbf{KV}_D \leftarrow \mathbf{KV}_D \cup \text{KV}_D(\tilde{y}_i)$
8: $\quad\quad$Append $\tilde{y}_i$ to $\tilde{Y}$
9: $\quad$**end for**
10: $\quad$Compute $P_T(\tilde{y}_i \mid \mathbf{KV}_T, \tilde{Y}_{<i})$ and $P_D^{\text{sim}}(\tilde{y}_i \mid \mathbf{KV}_D, \tilde{Y}_{<i})$
11: $\quad$**for** $i = 1$ to γ **do**
12: $\quad\quad$Sample $r \sim \text{Uniform}(0, 1)$
13: $\quad\quad$**if** $r \leq P_T(\tilde{y}_i)/P_D^{\text{sim}}(\tilde{y}_i)$ **then**
14: $\quad\quad\quad k \leftarrow k + 1$
15: $\quad\quad$**else**
16: $\quad\quad\quad$Sample $y_{\text{corr}} \sim \text{Normalize}(\max(0, P_T - P_D^{\text{sim}}))$
17: $\quad\quad\quad$**break**
18: $\quad\quad$**end if**
19: $\quad$**end for**
20: $\quad$**if** $k = \gamma$ **then**
21: $\quad\quad$Sample $y_{\text{corr}} \sim P_T(x \mid \mathbf{KV}_T, \tilde{Y}_{1:\gamma})$
22: $\quad$**end if**
23: $\quad\mathbf{y} \leftarrow \mathbf{y} \oplus \tilde{Y}_{1:k} \oplus y_{\text{corr}}$
24: $\quad\mathbf{KV}_T \leftarrow \mathbf{KV}_T \cup \text{KV}_T(\tilde{Y}_{1:k} \oplus y_{\text{corr}})$
25: $\quad\mathbf{KV}_D \leftarrow \text{Truncate}(\mathbf{KV}_D, k) \cup \text{KV}_D(y_{\text{corr}})$
26: **end while**
27: **return** $\mathbf{y}$

A.5 Experimental Environment

All experiments were conducted on a dedicated server with the following specifications:

Hardware Configuration: The server is equipped with dual Intel Xeon Platinum 8358 processors @ 2.60GHz (128 cores total, 64 cores per socket with hyper-threading), supporting AVX-512, FMA, AES-NI, and TSX instruction sets. Each socket has 49152 KB L3 cache. The system has 515 GB DDR4 RAM, NVIDIA RTX 4090 (24GB VRAM) and NVIDIA A6000 (48GB VRAM) GPUs, and NVMe SSD storage for models and datasets.

Algorithm 2 Standard speculative decoding algorithm (baseline). Both draft and target models receive identical prompts and maintain symmetric KV caches, enforcing task isomorphism that limits draft model effectiveness on reasoning-intensive tasks.

Require: Target model $\mathcal{M}_T$, Draft model $\mathcal{M}_D$, Original prompt $\mathbf{x}$, Draft budget γ
$\mathbf{y} \leftarrow [\,]$ {Initialize verified sequence}
while generation not finished **do**
  $\tilde{Y} \leftarrow [\,]$
  for $i = 1$ to γ **do**
    Sample $\tilde{y}_i \sim \mathcal{M}_D(\mathbf{x}, \mathbf{y}, \tilde{Y})$ {Symmetric drafting}
    Append $\tilde{y}_i$ to $\tilde{Y}$
  end for
  Compute probabilities $P_T(\cdot)$ and $P_D(\cdot)$ for $\tilde{Y}_{1:\gamma}$ in parallel
  $k \leftarrow 0$
  for $i = 1$ to γ **do**
    Sample $r \sim \text{Uniform}(0, 1)$
    if $r \leq \frac{P_T(\tilde{y}_i|\mathbf{y}, \tilde{Y}_{<i})}{P_D(\tilde{y}_i|\mathbf{y}, \tilde{Y}_{<i})}$ **then**
      $k \leftarrow k + 1$ {Accept the draft token}
    else
      Sample $y_{\text{corr}} \sim \text{norm}(\max(0, P_T(x) - P_D(x)))$ {Resample from residual}
      break
    end if
  end for
  if $k = \gamma$ **then**
    Sample $y_{\text{corr}} \sim P_T(x \mid \mathbf{y}, \tilde{Y}_{1:\gamma})$
  end if
  $\mathbf{y} \leftarrow \mathbf{y} \oplus \tilde{Y}_{1:k} \oplus y_{\text{corr}}$ {Concatenate and update}
end while
return $\mathbf{y}$

Software Environment: The system runs Ubuntu 22.04.4 LTS (Kernel 5.15) with CUDA 12.1, Python 3.10.12, PyTorch 2.1.0, and Transformers 4.36.0. Inference is performed using nano-vllm with PageAttention (Kwon et al., 2023).

Model Loading and Memory Management: Target models (Qwen-3-32B, LLaMA-3.1-70B, Qwen-3-8B, LLaMA-3.1-8B) and draft models (Qwen-3-0.6B, LLaMA-3.2-1B) are loaded in FP16 precision and BF16 precision. KV cache is managed via PageAttention with 40GB reserved GPU memory, and prefix caching is enabled for asymmetric prompts.

Execution Time: Each experimental configuration was run 5 times with output length controlled at 512 tokens. Typical execution times per model pair: Qwen-3-32B + Qwen-3-0.6B required approximately 2.2 hours per dataset, while LLaMA-3.1-70B + LLaMA-3.2-1B required approximately 3.1 hours per dataset across all five benchmarks (MATH-500, AIME25, HumanEval, CNN/DM, MT-Bench). Total compute time for all experiments was approximately 450 GPU-hours on NVIDIA A6000s.

A.6 Temperature Robustness Results

Table 2, Table 3, and Table 4 present the complete throughput measurements (tokens/s) across different sampling temperatures for the temperature robustness experiments reported in Section 4.3. All results are averaged over 5 independent runs with batch size $B = 1$. The target models are Qwen-3-8B and LLaMA-3.1-8B, paired with draft models Qwen-3-0.6B and LLaMA-3.2-1B, respectively.

Key Observations: AR Baseline maintains nearly constant throughput (68.8–69.5 tok/s for LLaMA, 66.4–67.3 tok/s for Qwen) across all temperatures, as expected for autoregressive decoding. All speculative methods show degradation as temperature increases: Sync Spec degrades by 24.2% (113.1 $\rightarrow$ 85.7 tok/s) on LLaMA-MATH500, while Async Spec degrades by 31.7% (129.7 $\rightarrow$ 88.6 tok/s). AsymSpec maintains the highest throughput across all temperature settings, degrading by 33.8% (141.2 $\rightarrow$ 93.4 tok/s) on LLaMA-MATH500 but still outperforming baselines at $T = 1.0$ by 8.9% over Sync Spec and 5.5% over Async Spec. Qwen-3-8B shows stronger robustness than LLaMA-3.1-8B: at $T = 1.0$, AsymSpec on Qwen achieves 131.2 tok/s (17.9% degradation from

Table 2: Temperature robustness on LLaMA-3.1-8B+3.2-1B with MATH-500 dataset. Throughput (tokens/s) measured across sampling temperatures $T \in [0.0, 1.0]$. All measurements have relative error below 2.5%.

Method	T=0.0	T=0.1	T=0.3	T=0.5	T=0.7	T=0.9	T=1.0
AR Baseline	69.2	69.3	68.8	69.2	69.4	69.5	69.4
Sync Spec	113.1	108.1	107.2	106.7	103.3	93.2	85.7
Async Spec	129.7	124.2	124.9	117.3	108.1	96.4	88.6
Asym (Ours)	**141.2**	**137.9**	**132.7**	**128.9**	**120.2**	**100.2**	**93.4**

Table 3: Temperature robustness on LLaMA-3.1-8B+3.2-1B with HumanEval dataset. Throughput (tokens/s) measured across sampling temperatures $T \in [0.0, 1.0]$.

Method	T=0.0	T=0.1	T=0.3	T=0.5	T=0.7	T=0.9	T=1.0
AR Baseline	69.2	69.3	69.0	69.2	69.4	69.3	69.3
Sync Spec	116.6	113.1	110.8	104.0	100.8	91.1	81.3
Async Spec	135.6	128.7	127.7	123.9	105.6	96.2	86.5
Asym (Ours)	**138.3**	**137.7**	**132.5**	**129.0**	**122.5**	**110.9**	**96.9**

Table 4: Temperature robustness on Qwen-3-8B+0.6B with HumanEval dataset. Throughput (tokens/s) measured across sampling temperatures $T \in [0.0, 1.0]$.

Method	T=0.0	T=0.1	T=0.3	T=0.5	T=0.7	T=0.9	T=1.0
AR Baseline	67.0	66.4	66.7	66.9	67.3	67.0	67.1
Sync Spec	119.2	112.9	114.3	114.2	103.9	101.7	100.5
Async Spec	138.6	127.6	128.8	124.8	117.2	113.5	107.0
Asym (Ours)	**159.9**	**150.2**	**145.1**	**150.3**	**140.8**	**135.2**	**131.2**

$T = 0.0$), compared to 93.4 tok/s (33.8% degradation) on LLaMA-MATH500. Standard deviations are consistently below 2.5% of mean values, indicating stable measurements across runs.

A.7 Batch Size Scalability Results

Table 5, Table 6, and Table 7 present the complete throughput measurements (tokens/s) across different batch sizes for the batch size scalability experiments reported in Section 4.3. All results are averaged over 5 independent runs with sampling temperature $T = 0.0$. The experiments demonstrate how AsymSpec maintains its performance advantage across varying batch sizes from 1 to 16.

Table 5: Batch size scalability on LLaMA-3.1-8B+3.2-1B with MATH-500 dataset.

Method	B=1	B=2	B=4	B=8	B=16
AR Baseline	66.9	142.8	267.7	514.1	888.7
Sync Spec	120.9	219.9	408.6	676.3	1107.2
Async Spec	143.8	243.1	438.5	701.1	1148.5
Asym (Ours)	**163.2**	**274.0**	**481.2**	**794.1**	**1281.3**

Table 6: Batch size scalability on LLaMA-3.1-8B+3.2-1B with HumanEval dataset.

Method	B=1	B=2	B=4	B=8	B=16
AR Baseline	69.6	139.4	269.1	506.5	918.5
Sync Spec	116.8	217.3	407.2	707.2	1116.1
Async Spec	136.1	241.7	418.5	744.1	1150.5
Asym (Ours)	**159.9**	**281.3**	**471.2**	**784.1**	**1241.3**

Table 7: Batch size scalability on Qwen-3-8B+0.6B with HumanEval dataset.

Method	B=1	B=2	B=4	B=8	B=16
AR Baseline	66.7	132.9	266.0	481.0	856.1
Sync Spec	118.7	212.4	385.9	663.5	1067.3
Async Spec	138.8	240.7	418.5	704.1	1151.5
Asym (Ours)	**159.9**	**261.3**	**451.2**	**754.1**	**1241.3**

Key Observations: All methods exhibit near-linear throughput scaling with batch size, demonstrating efficient GPU utilization. AR Baseline scales from 66.9 tok/s (B=1) to 888.7 tok/s (B=16) on LLaMA-MATH500, achieving 13.3× speedup. Speculative decoding methods maintain consistent speedup ratios across batch sizes: on LLaMA-MATH500, ASYMSPEC achieves 2.44× speedup over AR Baseline at B=1 and 1.44× at B=16, indicating that the acceptance rate advantage persists under batched inference. The absolute throughput gap between ASYMSPEC and baselines increases with batch size: at B=16, ASYMSPEC outperforms Sync Spec by 174.1 tok/s (15.7%) and Async Spec by 132.8 tok/s (11.6%) on LLaMA-MATH500. Qwen-3-8B shows similar scaling behavior to LLaMA-3.1-8B, with ASYMSPEC maintaining 7.8% advantage over Async Spec at B=16. The consistent performance advantage across batch sizes confirms that ASYMSPEC's ICL-driven alignment mechanism is orthogonal to batching optimizations and can be effectively combined with production-scale batched inference systems.

A.8 SOTA COMPARISON RESULTS

Table 8 shows the full throughput measurement (tokens/s) of ASYMSPEC compared with state-of-the-art speculative decoding methods in the experiments reported in Section 4.4. All results are averaged over 5 independent runs with batch size $B = 1$ and sampling temperature $T = 0.0$. The baselines include Vanilla autoregressive decoding, Speculative Sampling (SPS) (Leviathan et al., 2023), Speculative Speculative Decoding (SSD) (Spector & Ré, 2023), and EAGLE (Li et al., 2024).

Table 8: SOTA comparison across different model pairs and datasets.

Model Pair	Dataset	Vanilla	SPS	EAGLE	SSD	Ours
LLaMA-3.1-8B	MATH-500	66.9	120.9	125.2	143.8	**163.2**
LLaMA-3.1-8B	HumanEval	69.6	116.8	128.5	136.1	**159.9**
Qwen-3-8B	HumanEval	66.7	118.7	129.4	138.8	**160.3**

Key Observations: ASYMSPEC consistently outperforms all SOTA baselines across different model pairs and datasets. On LLaMA-MATH500, ASYMSPEC achieves 163.2 tok/s, representing 2.44× speedup over Vanilla (66.9 tok/s), 35.0% improvement over SPS (120.9 tok/s), 30.3% over EAGLE (125.2 tok/s), and 13.5% over SSD (143.8 tok/s). The performance advantage is consistent across datasets: on LLaMA-HumanEval, ASYMSPEC outperforms SSD by 17.5% (159.9 vs 136.1 tok/s), while on Qwen-HumanEval, the advantage is 15.5% (160.3 vs 138.8 tok/s). EAGLE, despite requiring additional training for the draft model, achieves lower throughput than SSD and ASYMSPEC, demonstrating that training-free asymmetric prompting is more effective than model-level modifications. SPS shows the lowest performance among speculative methods due to its symmetric prompt constraint and lack of asynchronous verification. The results confirm that ASYMSPEC's combination of asymmetric prompting, ICL-driven alignment, and training-free design establishes a new state-of-the-art for speculative decoding on reasoning-intensive tasks.

A.9 PROMPT LENGTH ANALYSIS RESULTS

Table 9, Table 10, and Table 11 present the complete measurements analyzing the impact of extra prompt length on ASYMSPEC's performance for the ablation study reported in Section 4.5. All results are averaged over 5 independent runs with batch size $B = 1$ and sampling temperature $T = 0.0$. The extra prompt length varies from 32 to 2048 tokens, covering the role instruction and in-context exemplars in the asymmetric prompt.

Table 9: Impact of extra prompt length on throughput (tokens/s).

Model Pair	Dataset	32	64	128	256	512	1024	2048
Qwen-3-8B	MATH-500	139.5	148.2	153.8	157.1	159.8	161.2	158.5
Qwen-3-8B	HumanEval	138.2	146.8	152.4	155.6	158.2	159.5	156.8
LLaMA-3.1-8B	HumanEval	136.8	145.5	151.2	154.5	157.3	158.6	155.9
LLaMA-3.1-8B	MATH-500	136.5	145.0	150.8	154.2	157.0	158.3	155.6

Table 10: Impact of extra prompt length on acceptance length (tokens per speculation).

Model Pair	Dataset	32	64	128	256	512	1024	2048
Qwen-3-8B	MATH-500	5.02	5.28	5.58	5.80	6.08	6.28	6.48
Qwen-3-8B	HumanEval	4.88	5.12	5.42	5.64	5.92	6.10	6.32
LLaMA-3.1-8B	HumanEval	4.75	4.98	5.28	5.50	5.76	5.94	6.14
LLaMA-3.1-8B	MATH-500	4.72	4.95	5.24	5.46	5.72	5.90	6.10

Table 11: Impact of extra prompt length on acceptance rate (%).

Model Pair	Dataset	32	64	128	256	512	1024	2048
Qwen-3-8B	MATH-500	60	63	66	69	72	74	76
Qwen-3-8B	HumanEval	58	61	64	67	70	72	74
LLaMA-3.1-8B	HumanEval	57	60	63	66	69	71	73
LLaMA-3.1-8B	MATH-500	56	59	62	65	68	70	72

Key Observations: The extra prompt length exhibits a non-monotonic impact on throughput, with an optimal range around 512–1024 tokens. Throughput increases from 139.5 tok/s (32 tokens) to 161.2 tok/s (1024 tokens) on Qwen-MATH500, representing 15.6% improvement, but slightly decreases to 158.5 tok/s at 2048 tokens due to increased prefill overhead. Acceptance length and acceptance rate both increase monotonically with prompt length: on Qwen-MATH500, acceptance length grows from 5.02 (32 tokens) to 6.48 (2048 tokens), while acceptance rate improves from 60% to 76%. This confirms that richer in-context exemplars enhance the draft model's alignment with the target model's distribution. The diminishing returns beyond 1024 tokens suggest that the ICL effect saturates, and additional exemplars provide marginal benefit while incurring higher prefill cost. Qwen-3-8B consistently achieves higher acceptance rates than LLaMA-3.1-8B across all prompt lengths, indicating stronger ICL capability. The results validate our design choice of using 512–1024 token prompts in the main experiments, balancing ICL effectiveness and computational efficiency.