Training-Free Adaptive Speculative Decoding via Linguistic Priors

Jingyi Wang, Jiaqi Huang*†, Zunnan Xu, Jun Zhou,
Kehong Yuan, Xiang Qian†

Tsinghua University
huangjq23@mails.tsinghua.edu.cn, qian.xiang@sz.tsinghua.edu.cn

Abstract

Speculative decoding (SPD) has emerged as a promising technique to accelerate Large Language Model (LLM) inference. However, current approaches typically enforce a uniform verification standard, neglecting the inherent heterogeneity of natural language and failing to distinguish between semantically-rich content and structurally-predictable syntax. In this paper, we propose LinguaSpec, a training-free framework that leverages linguistic priors to enable adaptive drafting and verification. Specifically, we introduce: (1) a Static Linguistic Probe (SLP) to categorize tokens with zero latency; (2) Syntactic Normalized Surprisal (SNS) to calibrate uncertainty against category-specific entropy; and (3) a dual strategy of Syntactically-Guided Elastic Expansion and POS-Adaptive Deferred Verification to dynamically adjust drafting depth and verification rigor. By balancing semantic integrity with structural efficiency, LinguaSpec significantly accelerates inference without requiring additional training. Experimental results demonstrate its superior performance across diverse benchmarks.

1 Introduction

Large Language Models (LLMs) (Vaswani et al., 2017) have demonstrated impressive capabilities across diverse domains (Dubey et al., 2024; Anil et al., 2023; DeepSeek-AI, 2025). However, their reliance on autoregressive decoding entails substantial latency, which becomes increasingly pronounced as model size scales. This sequential generation process requires accessing all model parameters for each token, creating a critical bottleneck for practical deployment in latency-sensitive settings.

The primary paradigm for mitigating the high inference latency of LLMs is Speculative Decoding(SPD), which partitions the decoding process into a cyclic draft-then-verify execution model. In this framework, a lightweight draft model first generates a sequence of candidate tokens, which are subsequently verified in parallel by a significantly larger target model. While this mechanism enables the potential for multiple-token acceptance per step, its practical efficiency is heavily dictated by the alignment between the draft and target models.

However, standard SPD is fundamentally constrained by its homogeneous verification standards, which treat all tokens as functionally uniform regardless of their linguistic properties. By relying primarily on information-theoretic metrics, such one-size-fits-all entropy-based strategies fail to account for the distinct functional roles that various parts-of-speech (POS) play in sentence construction. For instance, high entropy in semantic core words (e.g., nouns) often signifies benign linguistic variety, whereas equivalent entropy levels in structural tokens (e.g., punctuation) may indicate a critical failure in syntactic logic. As draft models encounter these heterogeneous linguistic units, the lack of syntactic awareness becomes an increasingly inefficient bottleneck, leading to either erroneous rejections of valid semantic variants or unproductive drafting of structural errors. This highlights the necessity for a more nuanced, syntactically-aware allocation of computational resources that can distinguish between semantic flexibility and syntactic integrity.

In this work, we present LinguaSpec, a training-free framework that integrates linguistic priors into the speculative decoding pipeline. Unlike existing methods that rely solely on probability distributions, LinguaSpec explicitly leverages the syntactic structure of language to guide both drafting and verification. Our approach is driven by three core innovations.

- First, we introduce the **Static Linguistic Probe (SLP)**, a zero-latency mechanism en-

* Project leader.
† Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 21184–21194

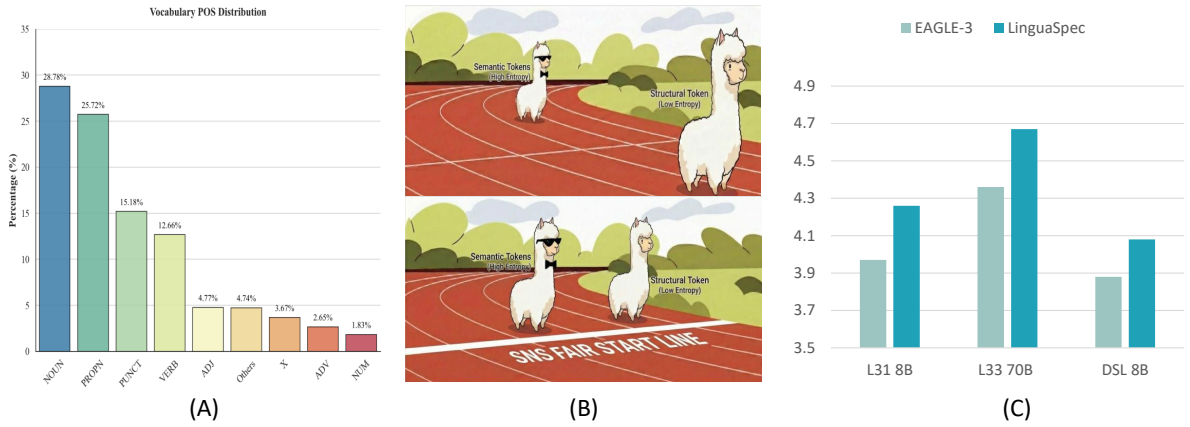

Figure 1: **Motivation and Performance of LinguaSpec.** (A) **Vocabulary Heterogeneity**: The POS distribution reveals that the vocabulary is composed of distinct roles, with Content words (Noun, Propn) dominating the distribution. (B) **Syntactic Fair Start Line**: We analogize verification to a race. Traditional methods (top) disadvantage high-entropy Content words. SNS (bottom) normalizes expected entropy, establishing a fair baseline where all tokens compete equally. (C) **Performance**: LinguaSpec achieves significant speedups over EAGLE-3 across multiple models.

abling $O(1)$ access. As shown in Figure 1, an analysis of the LLaMA vocabulary distribution reveals that the vocabulary is a heterogeneous entity composed of distinct roles rather than a homogeneous collection: Content categories (e.g., Nouns 28.78%, Proper Nouns 25.72%) form the semantic core, while Function categories exhibit significant structural characteristics, primarily consisting of Punctuation (15.18%) and other guiding words including particles and conjunctions (categorized as Others, 4.74%). SLP lays a solid foundation for subsequent differentiated processing by instantaneously identifying the category of each token.

- Second, we propose **Syntactic Normalized Surprisal (SNS)**, a metric that calibrates uncertainty based on the expected entropy of syntactic roles. As illustrated in Figure 1, we analogize the verification process to a race: under traditional mechanisms, tokens of different categories do not start from the same point. Content words, due to high information entropy, are naturally positioned "behind the starting line," whereas Function words, with extremely low entropy and high predictive certainty, are like runners with an inherent lead. SNS establishes a "Syntactic Fair Start Line" by normalizing the expected entropy of different categories to offset this innate difficulty disparity, allowing all tokens to compete on the same baseline. This ensures the system can accurately identify genuine model errors, avoiding the false rejection of drafts due to the inherent high-entropy nature of content words.

- Finally, we design a **Syntactically-Guided Elastic Expansion and POS-Adaptive Deferred Verification** mechanism, which achieves a synergistic optimization of generation depth and verification efficiency. When the SNS is low, the system dynamically increases the expansion depth of the Draft Tree. This increase in depth directly provides longer verifiable sequences on valid branches, thereby significantly enhancing the utilization of the deferred verification window during parallel processing. Simultaneously, the system enforces strict verification for content words to preserve semantic integrity, while applying relaxed verification for structural words to maximize efficiency, ultimately achieving an optimal balance between generation quality and inference speed.

2 Related Work

The foundational paradigm of Speculative Decoding (SPD) accelerates the autoregressive generation of Large Language Models (LLMs) by employing a lightweight drafter to propose candidate sequences, which are then verified in parallel by the target model (Xia et al., 2023; Chen et al., 2023;

Leviathan et al., 2023). Building on this structural foundation, existing research has branched into specialized alignment training, adaptive inference strategies, and architectural innovations.

2.1 Training-based SPD

A large body of research (Li et al., 2025a; Liu et al., 2024; Xiao et al., 2024; Ankner et al., 2024) extends SPD by learning auxiliary predictors to better align the drafter and target, thereby improving speculation accuracy. Typical strategies train additional modules that imitate the target's behavior to raise acceptance rates and speedups. Gumiho (Li et al., 2025a), Medusa (Cai et al., 2024), and ParallelSpec (Xiao et al., 2024) use hidden states from the base LLM as inputs to multiple lightweight MLP heads, each predicting a future token. EAGLE (Li et al., 2024b) generalizes this design by employing lightweight Transformer predictors and concatenated token–state pairs, while EAGLE-2 (Li et al., 2024a) improves efficiency with a dynamic tree–based candidate selection mechanism. EAGLE-3 (Li et al., 2025d) further leverages intermediate hidden states to scale up decoding acceleration. Beyond architectural modifications, several works incorporate trained modules for fine-grained control: SpecDec++ (Huang et al., 2024) and AdaEAGLE (Zhang et al., 2024b) predict early stopping or draft length, C2T (Huo et al., 2025) calibrates joint probability bias, and JudgeDecoding (Bachmann et al., 2025) employs an auxiliary classifier to determine contextual validity beyond exact matches. Although effective, these learning-based approaches typically rely on task-specific supervision and significant training costs.

2.2 Training-free SPD

In contrast, training-free approaches dispense with additional training by leveraging existing information or sub-modules for drafting. Retrieval-based strategies like REST (Fu et al., 2024) fetch tokens from external datastores, while Prompt Lookup Decoding exploits n-gram repetitions within the context. Parallel to this, layer-skipping techniques have evolved from static selection in Draft & Verify (Zhang et al., 2024a) to dynamic, context-aware routing in LayerSkip (Elhoushi et al., 2024). Further, FLy (Li et al., 2025b) and HeteroSpec (Liu et al., 2025) utilize entropy-based gates to dynamically adjust verification criteria or drafting depth.

2.3 Architectural and Structural Innovations

Beyond specific alignment strategies, another research line focuses on structural efficiency and resource reuse. To move beyond linear verification, tree-based attention mechanisms have been introduced to enable the simultaneous validation of multiple candidate branches (Miao et al., 2024). Furthermore, several studies maximize resource efficiency during the drafting stage: MoA (Zimmer et al., 2024) reuses target model KV caches, while other approaches leverage partial target model weights (Yi et al., 2024; Liu et al., 2024; Sun et al., 2024; Svirschevski et al., 2024) or share feature representations to bypass the need for independent draft models.

2.4 Linguistic Priors in Efficient Inference

The utilization of linguistic features, particularly Part-of-Speech (POS) tagging and syntactic dependencies, has long been a cornerstone in interpreting and steering Large Language Models (LLMs) (Tenney et al., 2019; Hewitt and Manning, 2019). While extensive research in linguistic probing has established that models implicitly encode rich hierarchical structures (Jawahar et al., 2019), these insights have been predominantly applied to enhance generation quality or enforce controllability (Zhou et al., 2023). Consequently, their potential utility in inference acceleration—a domain currently dominated by speculative sampling approaches (Leviathan et al., 2023; Chen et al., 2023)—remains largely under-explored. Although prior works in constrained decoding have attempted to integrate syntactic rules to guide generation (Shin et al., 2021; Roy et al., 2023), these methods typically rely on heavy auxiliary models or necessitate task-specific fine-tuning, thereby introducing significant computational overheads that often negate the efficiency gains sought in deployment scenarios.

3 Methodology

We propose LinguaSpec, a linguistically-adaptive framework designed to robustly accelerate LLM inference. The fundamental motivation of our work stems from the observation that tokens in natural language are not created equal: different Part-of-Speech (POS) categories play distinct roles in syntactic structure and semantic expression, yet existing speculative decoding methods treat them uniformly. This "one-size-fits-all" approach often leads to suboptimal verification, as it ignores the

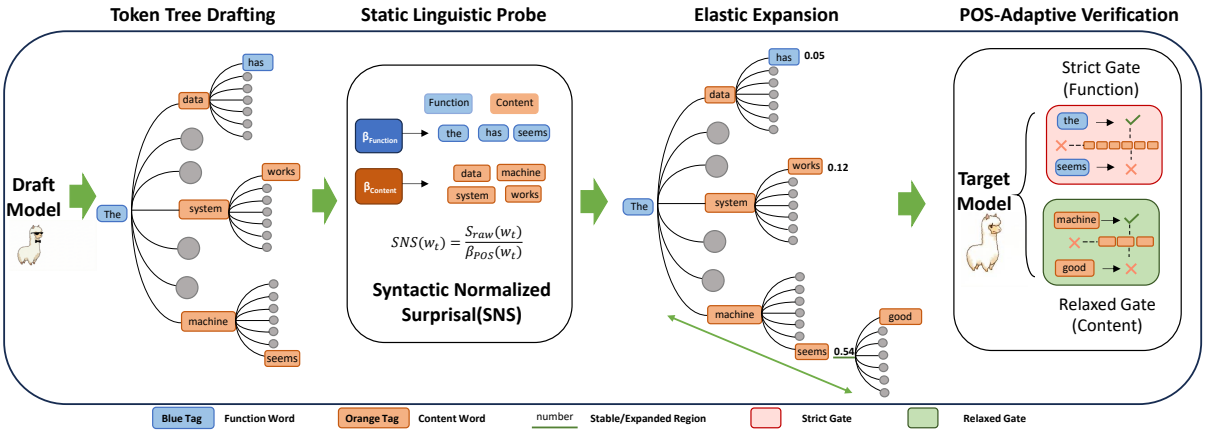

Figure 2: Overview of the LinguaSpec framework. The pipeline consists of: (1) **Static Linguistic Probe (SLP)** which assigns POS tags to tokens; (2) **Syntactic Normalized Surprisal (SNS)** for syntax-aware uncertainty estimation; (3) **Syntactically-Guided Elastic Expansion** that dynamically extends stable branches; and (4) **POS-Adaptive Deferred Verification** which applies strict constraints to function words and relaxed constraints to content words.

intrinsic linguistic properties that govern token generation. LinguaSpec addresses this by integrating static linguistic priors into the speculative drafting process. This section details the three core components of our framework: the Static Linguistic Probe for zero-latency syntactic tagging, the Syntactic Normalized Surprisal for fair uncertainty measurement, and the Syntactically-Guided Elastic Expansion.

3.1 Static Linguistic Probe

To strictly satisfy the "Training-free" constraint, our framework leverages the static nature of the LLM's vocabulary to embed linguistic priors. We construct a Static Linguistic Probe (SLP), which functions as a pre-computed mapping table $M : \mathcal{V} \to \mathcal{C}$. This mapping categorizes each token in the vocabulary $\mathcal{V}$ into one of four coarse-grained syntactic roles $\mathcal{C} = \{\text{Function, Content, Special, Other}\}$:

- **Function**: Structural tokens (e.g., determiners, prepositions) that establish grammatical structure.
- **Content**: Semantic tokens (e.g., nouns, verbs) that carry the primary information.
- **Special**: Formatting tokens including punctuation and code keywords.
- **Other**: Subwords or tokens outside standard linguistic boundaries.

By pre-computing this mapping offline, SLP enables $O(1)$ access to linguistic features during inference, incurring negligible latency.

3.2 Syntactic Normalized Surprisal

To distinguish between model confusion and natural semantic diversity, we introduce a normalized metric that contextualizes uncertainty. We first define the raw surprisal of a token w_t generated by the draft model as the negative log-likelihood of its probability:

$$S_{raw}(w_t) = -\ln P_{draft}(w_t|w_{<t}) \quad (1)$$

However, using raw logits or surprisal to measure uncertainty across different POS categories is inherently unfair. Content words often have multiple valid synonyms, leading to a naturally dispersed probability distribution (higher entropy) without necessarily indicating model error. In contrast, function words are often syntactically constrained with fewer valid alternatives, where high entropy typically signals a structural error.

To quantify this distinction, we introduce the Syntactic Expected Entropy (β_c), which represents the statistical average surprisal for a given Part-of-Speech (POS) category c derived from a general corpus. This baseline captures the intrinsic uncertainty associated with different syntactic roles. Leveraging these baselines, we define the Syntactic Normalized Surprisal (SNS) as the ratio of the raw surprisal to the expected entropy of the token's category:

$$\text{SNS}(w_t) = \frac{S_{raw}(w_t)}{\beta_{POS(w_t)}} \quad (2)$$

This metric functions as a syntax-aware normalization mechanism. By calibrating the raw surprisal

against the expected entropy of the corresponding POS category, SNS eliminates the inherent distributional bias between different linguistic roles. This allows for a unified verification standard where the uncertainty of a token is evaluated relative to its syntactic norm, rather than its absolute probability. Consequently, it prevents the system from being overly sensitive to the naturally high entropy of content words while maintaining strictness for structurally constrained function words.

3.3 Syntactically-Guided Elastic Expansion

We extend the static tree depth configuration of EAGLE-3 (Li et al., 2025d) with a dynamic expansion mechanism to further optimize the drafting process, particularly addressing the verification under-utilization problem in high-entropy contexts.

Standard speculative decoding methods typically rely on a fixed depth hyperparameter. This static approach often leads to suboptimal resource utilization: a shallow tree under-utilizes the target model's parallel verification bandwidth, while an excessively deep tree wastes computation on low-quality candidates. To address this trade-off, we propose Syntactically-Guided Elastic Expansion, a mechanism that dynamically adjusts the draft depth based on real-time generation stability.

The drafting process initiates with a standard expansion up to a pre-defined base depth D. Upon reaching this boundary, we introduce a Stability Assessment step to determine if the generation momentum warrants continuation. We posit that if the draft model maintains high confidence at the boundary of the original depth, it indicates a stable generation state—such as the production of a common idiom or a predictable syntactic structure. In such cases, truncating the generation is premature.

To quantify this, we evaluate the syntactic stability of the current leaf nodes. If the maximum normalized confidence score (derived from SNS, denoted as $\mathcal{S}$) among the active branches exceeds a threshold γ, the system triggers an Extension Phase. During this phase, the draft model continues to expand for an additional K steps. This conditional extension is formally defined as:

$$D_{final} = \begin{cases} D + K & \text{if } \max(\mathcal{S}_{leaves}) > \gamma \\ D & \text{otherwise} \end{cases} \quad (3)$$

By elastically extending the depth only when the model exhibits high syntactic confidence, LinguaSpec captures longer valid sequences within a single verification cycle. Crucially, because this decision is guided by our normalized SNS metric rather than raw probability, it avoids aggressive exploration in genuinely uncertain contexts. This strategy effectively amortizes the fixed cost of target model verification, ensuring that additional computational effort is invested only when it is statistically likely to yield higher acceptance rates.

3.4 POS-Adaptive Deferred Verification

Standard verification mechanisms, such as exact matching, enforce a rigid acceptance policy. This often leads to inefficiency, as valid drafts conveying the same semantic meaning but using different tokens are rejected. To address this, recent approaches (Li et al., 2025c) have introduced deferred verification, which tentatively accepts a mismatched token if its probability under the target model exceeds a certain threshold. The verification then continues for a subsequent window of tokens; if consistency is regained, the initial deviation is accepted.

However, these methods face a critical trade-off between performance and accuracy. A lenient configuration (e.g., low thresholds or short windows) increases the acceptance rate but risks admitting incorrect tokens, leading to significant generation deviation. Conversely, strict settings preserve accuracy but limit speedup gains. It is challenging to find a single static configuration that optimizes both metrics simultaneously.

We address this dilemma by integrating linguistic priors into the deferred verification framework. We observe that the tolerance for deviation varies significantly across different Part-of-Speech (POS) categories. Consequently, we propose a **POS-Adaptive Deferred Verification** strategy that assigns specific verification hyperparameters to different POS categories. While the configuration is fine-grained, it generally follows these principles:

- **Strict Constraints for Semantic Integrity**: For POS categories that carry core meaning (e.g., content words), we enforce **higher thresholds and longer verification windows**. Given that these tokens are the primary vehicles of information, any deviation requires scrutiny over a broader context to prevent semantic drift and ensure high generation accuracy.
- **Relaxed Constraints for Structural Efficiency**: For categories that are syntactically

localized (e.g., function words), we apply **lower thresholds and shorter verification windows**. Since structural errors typically manifest immediately within a short scope, a compact checking window is sufficient to capture them, thereby maximizing inference speed without redundant long-range verification.

- **Exception for Logical Negations**: While function words generally receive relaxed verification, high-impact logical negations (e.g., "not", "never", "cannot") represent a critical exception. Missing or hallucinating these tokens can trigger catastrophic semantic flips. Although such errors typically cause a stark divergence in the subsequent probability distribution—naturally triggering rejection under our standard mechanism—we introduce a deterministic failsafe. We explicitly hardcode this small, closed set of negations into the Strict Verification category. This zero-overhead refinement provides an absolute guarantee of semantic integrity in highly sensitive applications without sacrificing inference speed.

By dynamically adjusting the verification rigor based on linguistic roles, our method achieves a superior balance between inference acceleration and generation quality.

4 Experiments

4.1 Experimental Setup

Tasks. To ensure a fair comparison with the state-of-the-art baseline EAGLE-3, we aligned our task and model settings accordingly, adopting identical weights for all tasks without performing task-specific fine-tuning. Specifically, we evaluated multi-turn dialogue, code generation, mathematical reasoning, and instruction following using the public datasets MT-bench (Zheng et al., 2023), HumanEval, GSM8K (Cobbe et al., 2021), and Alpaca, respectively. Additionally, we incorporated ACPBench (Kokel et al., 2025) as an out-of-domain test set to assess the generalization capability of our method.

Models. We conduct experiments on LLaMA-3.1-8B-Instruct, LLaMA-3.3-70B-Instruct (Grattafiori et al., 2024), and DeepSeek-R1-Distill-LLaMA 8B. For the 8B models, experiments are conducted on 1×NVIDIA A100 80G GPU. For the 70B model, we use 2×A100 GPUs due to memory limitations.

Metrics. We evaluate the performance using the following metrics:

- **Speedup Ratio**: Actual acceleration compared to vanilla autoregressive decoding, measured by wall-clock time.
- **Average Acceptance Length** (τ): Mean tokens generated per drafting-verification cycle.
- **Quality Metrics**: Unlike standard speculative decoding which enforces exact matching, our LinguaSpec employs POS-Adaptive Deferred Verification which may introduce minor token deviations. Therefore, we report task-specific metrics (i.e., Accuracy for GSM8K and Pass@1 for HumanEval) to ensure that the semantic integrity is preserved.

Implementation Details. To construct the Static Linguistic Probe (SLP), we adopt a hierarchical, heuristic processing pipeline. For tokens that inherently exist as complete words, we tag them with their statistically dominant Part-of-Speech using NLTK and WordNet. For tokens that do not form complete words (e.g., BPE fragments), we intentionally do not attempt to assign them traditional semantic tags. If a long word is decomposed by the tokenizer, assigning independent semantic properties to its individual affixes or fragments would be linguistically inappropriate and introduce severe semantic confusion. Instead, we heuristically filter them using string formatting, such as the tokenizer's leading space mechanism, and statically categorize them as "Subword Fragments" under the 'Other' structural macro-category. In the context of a complete sentence, these sub-words function merely as structural continuations rather than independent semantic units, which strictly aligns with applying our relaxed verification strategy. This zero-overhead, rule-based pipeline ensures precise classification while maintaining $\mathcal{O}(1)$ access during inference.

For the Syntactic Normalized Surprisal (SNS), the expected entropy β_c was calibrated on the WikiText-2 dataset. The calculated baselines are $\beta_{func} \approx 2.04$, $\beta_{content} \approx 3.77$, and $\beta_{special} \approx 2.34$. Regarding hyperparameters, we set the base draft tree depth to 6. For the Elastic Expansion, the extension step K is set to 4, and the stability

		MT-bench		HumanEval		GSM8K		Alpaca		Acp		Mean	
Model	Method	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ
L31 8B	EAGLE-3	3.55x	6.07	4.31x	6.49	4.03x	6.23	4.23x	6.91	3.71x	5.79	3.97x	6.30
	LinguaSpec	**3.93x**	**6.91**	**4.46x**	**7.72**	**4.39x**	**6.99**	**4.56x**	**7.97**	**3.95x**	**6.57**	**4.26x**	**7.23**
L33 70B	EAGLE-3	4.13x	5.81	4.77x	6.39	4.55x	6.29	4.61x	6.88	3.72x	5.17	4.36x	6.11
	LinguaSpec	**4.43x**	**6.33**	**5.30x**	**7.52**	**4.73x**	**6.82**	**4.99x**	**8.02**	**3.88x**	**5.53**	**4.67x**	**6.84**
DSL 8B	EAGLE-3	3.72x	5.76	3.98x	6.34	4.46x	6.77	3.75x	5.53	3.47x	5.55	3.88x	5.99
	LinguaSpec	**3.81x**	**6.38**	**4.22x**	**7.02**	**4.90x**	**8.34**	**3.84x**	**5.87**	**3.61x**	**5.93**	**4.08x**	**6.71**

Table 1: Speedup ratios and average acceptance lengths τ of different methods. L31 represents LLaMA-Instruct 3.1, L33 represents LLaMA-Instruct 3.3, and DSL represents DeepSeek-R1-Distill-LLaMA. LinguaSpec consistently outperforms baselines across all tasks.

threshold γ is set to -1.0. In the POS-Adaptive Deferred Verification, we apply distinct configurations based on syntax. For Content words, we employ a probability threshold of 0.3 and a larger verification window ($W = 4$). Given that these tokens carry the core semantic meaning and are susceptible to complex errors, this configuration ensures robust verification with sufficient context. Conversely, for Function and Special tokens which are syntactically localized, we use a threshold of 0.2 and a smaller verification window ($W = 2$), premised on the intuition that structural errors can be rapidly identified within a short scope.

Model	Method	GSM8K (Acc.%)	HumanEval (Pass@1)
LLaMA-3-8B	Vanilla	82.71	69.51
	LinguaSpec	83.02	68.29
LLaMA-3-70B	Vanilla	96.13	78.66
	LinguaSpec	96.06	79.88
DSL 8B	Vanilla	73.77	57.32
	LinguaSpec	73.77	56.71

Table 2: Quality comparison between Vanilla autoregressive decoding and LinguaSpec. We report Accuracy (%) for GSM8K and Pass@1 (%) for HumanEval. The results demonstrate that LinguaSpec maintains or even slightly improves generation quality.

4.2 Main Results

We present the main experimental results in Table 1. Across all evaluated models and tasks, LinguaSpec consistently outperforms the baselines in terms of inference speed while maintaining high generation quality.

Speedup Performance. As shown in Table 1, LinguaSpec achieves state-of-the-art speedups across all models. On LLaMA-3.1-8B, it attains a mean speedup of $4.26\times$, significantly outperforming EAGLE-3 ($3.97\times$). Notably, on code generation tasks (HumanEval), LinguaSpec reaches a remarkable $4.46\times$ speedup. This advantage extends to larger models; on LLaMA-3.3-70B, we achieve a $4.67\times$ mean speedup compared to $4.36\times$ for EAGLE-3. Furthermore, on the out-of-domain ACPBench, LinguaSpec demonstrates superior generalization (e.g., $3.95\times$ vs. $3.71\times$ on LLaMA-3.1-8B), indicating that our linguistic priors remain effective even under distribution shifts where trained drafters may falter. Even on the distilled DeepSeek-R1 model, our method maintains a lead with $4.08\times$ speedup, demonstrating the robustness of our linguistic-guided approach across different training paradigms.

Quality Preservation. Table 2 presents the detailed performance comparison. LinguaSpec maintains high generation quality across all benchmarks. Notably, in some instances, LinguaSpec yields slightly higher scores than the vanilla baseline (e.g., +0.31% on GSM8K for 8B and +1.22% on HumanEval for 70B). We attribute this to the flexibility introduced by our POS-Adaptive Deferred Verification. Unlike rigid exact matching, our approach permits semantically equivalent but lexically distinct content words. This controlled diversity potentially allows the model to explore superior generation paths that might be suppressed by standard constraints, effectively acting as a syntax-aware local search that enhances expressivity without compromising correctness.

4.3 Ablation Studies

To verify the contribution of each component in LinguaSpec, we conduct ablation studies on the LLaMA-3-8B model on the GSM8K dataset.

Effect of Syntactic Normalized Surprisal (SNS). As shown in Table 3, replacing SNS with raw probability-based confidence estimation—which essentially simulates a uniform entropy threshold akin to Typical Sampling or the Typical Acceptance criteria used in pioneering works like Medusa (Cai et al., 2024)—causes the speedup to drop from

$4.39\times$ to $3.99\times$. While this uniform threshold still surpasses the EAGLE-3 baseline ($3.97\times$), it struggles to distinguish between naturally high-entropy content words and actual model uncertainty. By incorporating linguistic priors to establish category-specific expected entropy baselines (β_c), the full SNS significantly boosts the speedup up to $4.39\times$. This empirically demonstrates the distinct advantage of syntax-aware thresholds over uniform entropy criteria, effectively calibrating confidence to avoid inefficient drafting and yield a higher acceptance rate.

Impact of Syntactically-Guided Elastic Expansion. Removing the elastic expansion mechanism reduces the speedup to $4.18\times$. This demonstrates that dynamic tree depth is crucial for capitalizing on stable generation phases. By extending the draft tree only when syntactic confidence is high, our method captures longer valid sequences (higher τ) without wasting computation on unstable branches.

Efficacy of POS-Adaptive Deferred Verification. Our adaptive verification strategy also plays a vital role. Reverting to a strict exact-matching standard (i.e., removing deferred verification) reduces the speedup to $4.15\times$. This highlights that a rigid "exact match" policy is a major bottleneck. Our POS-adaptive approach unlocks this potential by allowing flexible verification for content words while maintaining structural integrity.

Method Variant	Speedup	τ
EAGLE-3 (Baseline)	4.03x	6.23
LinguaSpec (Full)	**4.39x**	**6.99**
w/o SNS (Raw Prob.)	3.99x	6.46
w/o Elastic Expansion	4.18x	6.44
w/o Deferred Verif. (Exact Match)	4.15x	6.64

Table 3: Ablation analysis of LinguaSpec components on LLaMA-3-8B (measured on GSM8K). Each component contributes significantly to the overall inference acceleration and acceptance length. Notably, all variants of LinguaSpec outperform the EAGLE-3 baseline ($3.97\times$).

4.4 Hyperparameter Sensitivity

We further investigate the impact of key hyperparameters on the performance of LinguaSpec.

Impact of Expansion Depth. The elastic expansion step size K determines how aggressively the draft tree grows when stability is detected. Table 4 shows that performance peaks at $K = 4$. While larger K yields a higher average acceptance length (τ) by capturing longer valid sequences, it concurrently imposes linear growth in drafting latency. This increased overhead eventually negates the benefits of additional token acceptance, resulting in diminished overall speedup. To isolate the benefit of our dynamic mechanism, we compared it against a static baseline where the draft tree depth is unconditionally increased by 4 (Static (+4)). Although this static approach yields a higher acceptance length ($\tau = 7.44$), the overall speedup drops to $3.93\times$. This is due to the uniformly higher drafting latency (54.6ms vs. 46.7ms for $K = 4$), which outweighs the benefits of longer drafts. LinguaSpec optimizes this trade-off by investing computation only when likely to be profitable.

Expansion Config.	Speedup	τ
Dynamic $K = 2$	4.17x	6.86
Dynamic $K = 4$	**4.39x**	**6.99**
Dynamic $K = 6$	4.25x	7.06
Dynamic $K = 8$	3.87x	7.07
Static (+4)	3.93x	7.44

Table 4: Impact of expansion strategy (measured on GSM8K). While statically increasing depth yields longer drafts (τ), our dynamic expansion ($K = 4$) achieves better speedup by avoiding unnecessary latency.

Stability Threshold γ	Speedup	τ
-0.05	4.23x	6.51
-1.0	**4.39x**	**6.99**
-2.0	4.10x	7.01
-3.0	3.89x	7.03

Table 5: Impact of stability threshold γ on speedup and acceptance length (measured on GSM8K, with $K = 4$). $\gamma = -1.0$ offers the best trade-off.

Impact of Stability Threshold. The threshold γ controls the sensitivity of the elastic expansion. As shown in Table 5, a strict threshold (e.g., $\gamma = -0.05$) limits the frequency of expansion, resulting in shorter acceptance lengths. Conversely, an overly loose threshold (e.g., $\gamma = -3.0$) triggers expansion too frequently even for unstable branches, increasing computational cost without proportional gain. We find that $\gamma = -1.0$ yields the best speedup by effectively identifying stable generation phases.

Impact of POS-Adaptive Configuration. We compare our adaptive strategy against uniform ver-

ification window settings. As illustrated in Table 6, increasing the uniform window size from 2 to 4 improves generation quality (Pass@1 rises from 65.1% to 68.4%) but comes at a significant cost to inference speed (speedup drops from 4.65× to 4.15×). This highlights the inherent conflict in static configurations. In contrast, our LinguaSpec employs a differentiated approach—applying strict verification ($W = 4$) only to semantic content while relaxing constraints ($W = 2$) for structural tokens. This allows us to achieve a speedup (4.46×) comparable to smaller windows while maintaining the robustness (68.3%) of larger windows, effectively resolving the trade-off.

Verification Strategy	Speedup	HumanEval Pass@1
Uniform $W = 2$	**4.65x**	65.1%
Uniform $W = 3$	4.40x	67.9%
Uniform $W = 4$	4.15x	**68.4%**
Adaptive (Ours)	<u>4.46x</u>	<u>68.3%</u>

Table 6: Comparison between Uniform Verification Windows and our POS-Adaptive Strategy. Bold indicates the absolute best value in each column, while underlines highlight the optimal trade-off achieved by our method, bridging the quality of large windows with the efficiency of small ones.

5 Conclusion

In this paper, we introduced LinguaSpec, a novel training-free framework that bridges the gap between linguistic structure and speculative decoding. By moving beyond rigid verification paradigms, we demonstrated that incorporating linguistic priors can significantly enhance inference efficiency. Our strategy dynamically adjusts drafting and verification rigor based on syntactic roles, ensuring optimal resource allocation. Extensive experiments confirm that LinguaSpec achieves state-of-the-art performance, delivering up to 5.30× speedup on LLaMA-3.3-70B-Instruct, while consistently preserving generation quality. Crucially, as a plug-and-play solution requiring no additional training, it offers a practical path for deploying large language models in latency-sensitive applications. Future work will explore finer-grained linguistic features to further refine the adaptive verification process.

6 Limitations

Despite the promising results, LinguaSpec has several limitations. First, the effectiveness of our Static Linguistic Probe (SLP) and Syntactic Normalized Surprisal (SNS) relies on the quality of the underlying linguistic priors. While we calibrated our metrics on general corpora (WikiText-2), these priors may not optimally align with highly specialized domains (e.g., biomedical text) or languages with significantly different syntactic structures, potentially requiring domain-specific recalibration. Second, our current approach employs coarse-grained POS categories to ensure zero-latency processing. This simplification may overlook subtle syntactic dependencies that could further refine the drafting process. Future iterations could explore lightweight, dynamic parsing to capture these finer-grained features. Third, while POS-Adaptive Deferred Verification effectively balances speed and quality, it introduces a non-zero risk of semantic drift, particularly in tasks demanding strict token-level exactness. Although our experiments show this impact is negligible, users with extreme precision requirements may need to tune the verification window conservatively. Finally, we acknowledge that quantitative metrics (e.g., Pass@1) might not fully capture subtle stylistic nuances. However, following the fundamental premise of speculative decoding, the target model serves as the absolute oracle. LinguaSpec aims solely to accelerate the target model's inherent probability distribution, not to alter its stylistic or semantic choices. In the worst-case scenario where a stylistically plausible draft is rejected, the system simply falls back to the target model's natural generation, guaranteeing zero degradation from the baseline capabilities and preserving a clear boundary between inference acceleration and stylistic alignment.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

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 Kohler. 2025. Judge decoding: Faster speculative sampling requires going beyond model alignment. *arXiv preprint arXiv:2501.19309*.

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. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Charlie Chen and 1 others. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Keshwam, Ahmad Al-dahle, Adithya Renduchintala, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Koumchatzky, and Jelena Yangel. 2024. Layerskip: 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*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4129–4138.

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

Feiye Huo, Jianchao Tan, Kefeng Zhang, Xunliang Cai, and Shengli Sun. 2025. C2t: A classifier-based tree construction method in speculative decoding. *arXiv preprint arXiv:2502.13652*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. 2025. Acpbench: Reasoning about action, change, and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26559–26568.

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

Jinze Li, Yixing Xu, Haiduo Huang, Xuanwu Yin, Dong Li, Edith CH Ngai, and Emad Barsoum. 2025a. Gumiho: A hybrid architecture to prioritize early tokens in speculative decoding. *arXiv preprint arXiv:2503.10135*.

Jinze Li, Yixing Xu, Guanchen Li, Shuo Yang, Jinfeng Xu, Xuanwu Yin, Dong Li, Edith CH Ngai, and Emad Barsoum. 2025b. Training-free loosely speculative decoding: Accepting semantically correct drafts beyond exact match. *arXiv preprint arXiv:2511.22972*.

Jinze Li, Yixing Xu, Guanchen Li, Shuo Yang, Jinfeng Xu, Xuanwu Yin, Dong Li, Edith CH Ngai, and Emad Barsoum. 2025c. Training-free loosely speculative decoding: Accepting semantically correct drafts beyond exact match. *arXiv preprint arXiv:2511.22972*.

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

Yuhui Li, Fangyun Wei, Chamara Zhang, and Hongyang Zhang. 2024b. Eagle: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025d. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. 2024. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*.

Siran Liu, Yang Ye, Qianchao Zhu, Zane Cao, and Yongchao He. 2025. Heterospec: Leveraging contextual heterogeneity for efficient speculative decoding. *arXiv preprint arXiv:2505.13254*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, and 1 others. 2024. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949.

Shuo Roy and 1 others. 2023. Recode: Robustness evaluation of code generation models. *arXiv preprint arXiv:2212.10264*.

Richard Shin, Christopher H Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7699–7715.

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

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. *Advances in Neural Information Processing Systems*, 37:16342–16368.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

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.

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*.

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.

Jun Zhang, Jue Wang, Huan Li, Ladan Ding, Lianli Wu, and Gholamreza Haffari. 2024a. Draft & verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Preprint version appeared in 2023.

Situo Zhang, Hankun Wang, Da Ma, Zichen Zhu, Lu Chen, Kunyao Lan, and Kai Yu. 2024b. Adaeagle: Optimizing speculative decoding via explicit modeling of adaptive draft structures. *arXiv preprint arXiv:2412.18910*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Wangchunshu Zhou and 1 others. 2023. Controlled text generation with natural language instructions. *arXiv preprint arXiv:2304.13779*.

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