UniVer: A Unified Perspective for Multi-step and Multi-draft Speculative Decoding

Yepeng Weng[1,2*] **Qiao Hu**[3*†] **Takehisa Yairi**[1]
[1] The University of Tokyo [2]Lenovo AI Technology Center
[3] National Center for Mathematics and Interdisciplinary Sciences (NCMIS), AMSS, CAS
yweng@g.ecc.u-tokyo.ac.jp, huqiao2020@amss.ac.cn

Abstract

Speculative decoding accelerates Large Language Models via draft-then-verify, where verification can be framed as an Optimal Transport (OT) problem. Existing approaches typically handle multi-draft and multi-step aspects in isolation, applying either flat OT to single-step drafts or per-token rejection sampling to tree-structured candidates. This separation leaves the joint regime (where multi-step dependencies meet multi-draft branching) poorly optimized, as local verification rules fail to exploit the coupling between horizontal and vertical dimensions of candidate trees. In this paper, we propose a unified perspective that casts tree-based verification as a conditional OT problem. Our key insight is that vertical dependencies can be abstracted through prefix acceptance probabilities, which act as dynamic scaling factors to actively guide horizontal draft selection. Based on this principle, we introduce UniVer, a verification algorithm that jointly optimizes across tree levels by composing local optimal transport plans under prefix constraints. We prove that UniVer remains lossless and achieves the optimal acceptance rate under the proposed conditional framework. Extensive experiments across different tasks and models demonstrate that UniVer improves acceptance length by 4.2% to 8.5% over standard recursive rejection sampling without replacement and by up to 2.5% over the single-step OT-based Greedy method, while preserving the target distribution exactly.

1 Introduction

Modern Large Language Models (LLMs) suffer from high inference latency due to their autoregressive nature. Speculative decoding [13, 3] mitigates this problem without compromising output quality: a lightweight draft model predicts future tokens, which are then verified by the target model in parallel. A fundamental algorithmic challenge is to design an optimal verification strategy that maximizes the expected acceptance rate while maintaining statistical fidelity to the original model (*i.e.,* the output distribution is identical to that of vanilla autoregressive decoding).

Recent studies improve acceptance rates by extending the naive (local) verification strategy along two orthogonal dimensions: optimizing horizontal selection among multiple candidates and managing vertical dependencies across multiple steps. On one hand, multi-draft methods such as SpecTr [21] and Greedy method [8] select multiple draft tokens at each step and optimize the acceptance rates from an Optimal Transport (OT) perspective. However, these approaches remain vertically "myopic", treating each generation step as an independent event and failing to account for the sequential dependencies in tree-structured drafts. On the other hand, multi-step methods [20, 24] focus on optimizing verification vertically across sequences or trees. Yet, these methods are horizontally limited: they rely on local,

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

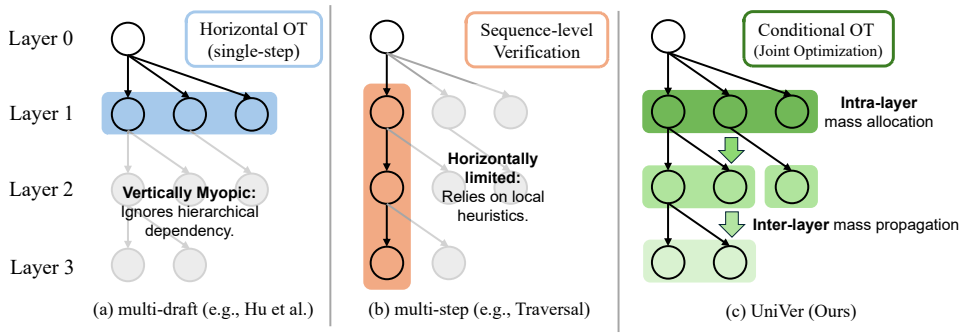


Figure 1: Conceptual illustration of verification paradigms. **(a) Multi-Draft approaches** (e.g., Hu et al. [8]) optimize horizontal selection via OT but ignore vertical dependencies (*vertically myopic*). **(b) Multi-Step approaches** (e.g., Traversal Verification [24]) optimize vertical acceptance but rely on local heuristics for horizontal selection (*horizontally limited*). **(c) UniVer** unifies both dimensions via Conditional OT, enabling parallel intra-layer mass allocation for different branches and seamless inter-layer mass propagation.

per-token rejection sampling, which is suboptimal compared to coordinated, multi-draft selection. Consequently, existing paradigms suffer from a structural trade-off: they either optimize across multiple candidates at a single depth or across multiple steps for a single candidate, but fail to achieve joint optimality across both dimensions (Figure 1).

This discrepancy raises a fundamental question: *How to unify multi-draft selection and multi-step coordination into a single, principled verification framework?* The core difficulty lies in the combinatorial complexity of a candidate tree, where the exponential growth of potential paths makes direct joint optimal transport highly non-trivial. To bypass this complexity, we shift the focus from solving the global OT problem to a recursive local-global coordination. By introducing the *prefix acceptance probability* of a node v ($\tilde{p}_v$) as the dynamic scaling factor, we reformulate horizontal verification into a conditional optimal transport problem. This enables a unified strategy that couples multi-draft selection with cross-level dependencies while maintaining strict losslessness.

Guided by this principle, we propose UniVer (Unified Verification), a principled framework that decomposes tree-structured verification into a top-down allocation phase and a post-order decision phase. Unlike prior approaches [15, 24] that rely on Recursive Rejection Sampling without replacement, UniVer enables parallel computation of acceptance plans for all branches at the same depth. By leveraging conditional OT, our algorithm determines the acceptance mass of child nodes within a single probability space, effectively eliminating the sequential dependency inherent in RRSw. As a result, UniVer provides a lossless yet efficient paradigm for tree speculative decoding.

Our contributions are summarized as follows:

- **Unified Framework.** We propose UniVer, a framework that unifies multi-step and multi-draft speculative decoding. By treating prefix acceptance probabilities $\tilde{p}_v$ as dynamic scaling factors, we cast tree-based verification as a series of conditional OT problems. The framework is not tied to a specific horizontal verifier and accommodates other single-step verification schemes as plug-in components.
- **Theoretical Foundation.** We prove that UniVer achieves the conditional optimum for any $\tilde{p}_v \in (0, 1]$ under a specific sampling strategy (*i.e.,* Greedy selection [8]), attaining the theoretical upper bound of acceptance mass under prefix constraints. We further show that this bound is strictly non-inferior to that of the Greedy method, establishing UniVer as a principled generalization of single-step OT optimality to multi-step tree verification.
- **Empirical Validation.** UniVer enables single-pass verification by eliminating the sequential dependencies inherent in RRSw and Traversal. Experiments across various LLMs and tree architectures show that UniVer consistently improves acceptance length by 4.2% to 8.5% over standard RRSw and by up to 2.5% over Greedy under temperature 1.0 while maintaining losslessness.

2 Preliminaries

2.1 Speculative Decoding and OT

Let $\{\mathcal{M}_b, \mathcal{M}_s\}$ be a pair of probabilistic models over vocabulary Σ. For any given accepted prefix $\mathbf{x}$, we refer to $\mathcal{M}_b(\cdot \mid \mathbf{x})$ as the target distribution and $\mathcal{M}_s(\cdot \mid \mathbf{x})$ as the draft distribution, respectively.

Single-Draft Case. In standard speculative decoding [13, 3], a single draft token $x \sim \mathcal{M}_s(\cdot \mid \mathbf{x})$ is verified against the target distribution $\mathcal{M}_b$. The acceptance rate is determined by the probability of alignment between $\mathcal{M}_s$ and $\mathcal{M}_b$. Formally, this corresponds to an optimal transport problem, where the optimal acceptance rate α^* equals one minus the total variation distance between $\mathcal{M}_b$ and $\mathcal{M}_s$ (*i.e.,* the probability mass shared by the target and draft distributions):

$$\alpha^* = \sum_{x \in \Sigma} \min\{\mathcal{M}_b(x \mid \mathbf{x}), \mathcal{M}_s(x \mid \mathbf{x})\}. \tag{1}$$

Local Heuristics for Multi-Draft. When n draft tokens are generated, a straightforward extension is to apply the single-token verification sequentially. Methods such as Recursive Rejection Sampling (RRS) and its without replacement variant (RRSw) [17, 4, 9, 27] adopt this strategy: they verify candidates one by one using local pairwise comparisons between $\mathcal{M}_b$ and $\mathcal{M}_s$, re-normalizing the residual distribution after each rejection. While RRSw prevents redundant sampling of identical tokens, it remains a *local heuristic* that processes candidates independently rather than coordinating their acceptance globally.

Multi-Draft with Optimal Transport. In contrast, some recent works [21, 8, 11] frame multi-draft verification as an OT problem. Hu et al. [8] propose a Greedy method. They first deterministically select $m-1$ drafts with highest probabilities from $\mathcal{M}_s$, then sample the final draft from the residual $\mathcal{M}_s^{\neg}$. Under this sampling strategy, they derive a closed-form verification strategy that achieves the theoretical upper bound of the acceptance rate:

$$\alpha^*_{\text{Greedy}}(\mathcal{M}_b, \mathcal{M}_s) = \sum_{i \in \text{Top}_{m-1}(\mathcal{M}_s)} \mathcal{M}_b(i) + \sum_{i \in \Sigma} \min\{\mathcal{M}_b(i), \mathcal{M}_s^{\neg}(i)\}, \tag{2}$$

where $\mathcal{M}_s^{\neg}$ denotes the residual draft distribution. Unlike RRSw, this OT-based approach works under the above sampling structure (Top_{m-1} plus one residual sample) and coordinates the acceptance of all m candidates in a unified probability space. Under such sampling scheme, Eq. (2) maximizes the marginal acceptance probability at each step.

2.2 Multi-step Strategies

Consider a draft model generating a candidate tree T. For clarity, we use $\mathbf{v} = (v_0, \ldots, v_\gamma)$ to represent a root-to-node chain $\mathbf{v} \subseteq T$, where v_0 denotes the root node and the chain length is $\gamma + 1$.

Local Heuristics. The baseline approach applies vanilla verification at each step independently. For any fixed chain $\mathbf{v} = (v_0, \ldots, v_\gamma)$, let the acceptance ratio be $r(v_i) = \mathcal{M}_b(v_i)/\mathcal{M}_s(v_i)$ at position v_i. Under local heuristics, the acceptance probability is computed independently at each position as $\min\{r(v_i), 1\}$. Consequently, the actual acceptance probability $p_{v_k}^{\text{Local}}$ of each prefix chain of $\mathbf{v}$ is given by the product of individual, locally truncated probabilities:

$$p_{v_k}^{\text{Local}} = \prod_{i=1}^{k} \min\{r(v_i), 1\},\ 1 \le k \le \gamma.$$

This *post-hoc aggregation* treats each verification step as independent, failing to exploit the coupling of probability mass along the sequence, which leads to suboptimal cumulative acceptance rates.

Block Verification. For single-chain decoding, Sun et al. [20] propose Block Verification. Unlike local verification, which independently truncates the local density ratio at 1.0 before accumulation, Block Verification computes the prefix acceptance probability $p_{v_k}^{\text{Block}}$ by first performing cumulative multiplication and then applying truncation:

$$p_{v_k}^{\text{Block}} = \min\left\{ r(v_k) \cdot p_{v_{k-1}}^{\text{Block}}, 1 \right\},\ 1 \le k \le \gamma,$$

where $p_{v_0}^{\text{Block}} = 1$. This mechanism enables a high density ratio ($r(v_k) > 1$) to recharge the prefix probability diminished by prior steps. In contrast, local verification follows $p_{v_k}^{\text{Local}} = \min\{r(v_k), 1\} \cdot p_{v_{k-1}}^{\text{Local}}$, which permanently penalizes the chain for any local mismatch. Consequently, Block Verification yields a strictly non-inferior cumulative acceptance probability, as $p_{v_k}^{\text{Block}} \geq p_{v_k}^{\text{Local}}$ holds for all $1 \leq k \leq \gamma$.

Traversal Verification. Traversal Verification [24] extend Block Verification to tree structures through a *post-order traversal* strategy. This approach leverages the joint probability for vertical optimization, but handles the horizontal dimension via RRSw. Specifically, Traversal Verification processes the tree bottom-up by iteratively verifying candidates against the residual target distribution, falling back to sibling nodes or the parent upon rejection. In other words, it lacks joint optimization across siblings, failing to exploit the horizontal coupling.

3 Method

3.1 Overview

To bridge the gap between local horizontal selection and suboptimal vertical coordination in existing methods, we present UniVer, a unified framework for multi-draft and multi-step speculative decoding. As presented in Figure 2, UniVer operates through a two-stage verification process that jointly optimizes horizontal and vertical optimization.

Allocation Phase: Top-Down Probability Propagation. Starting from the root node with initial probability $\tilde{p}_{root} = 1$, we propagate the acceptance mass layer by layer. For each layer, the acceptance plans for all independent branches are computed in parallel. For each node v, we treat $\tilde{p}_v$ as the *effective prefix acceptance probability* and compute the conditional optimal transport plan between the scaled target distribution $\tilde{p}_v \cdot \mathcal{M}_b(\cdot \mid v)$ and the draft distribution $\mathcal{M}_s(\cdot \mid v)$. This yields marginal acceptance probabilities $p_v(u)$ for each child $u \in \mathcal{C}(v)$, which are then normalized into conditional acceptance probabilities $\tilde{p}_u$ for the next layer. This propagation continues until all leaf nodes are processed.

Decision Phase: Post-Order Tree Traversal. With all acceptance probabilities precomputed, we perform a standard post-order tree traversal (as defined in Traversal Verification [24]). For each node $v \in T$, we draw $\eta \sim U(0, 1)$ and accept v if $\eta < \tilde{p}_v$ (leaf) or $\eta < \tilde{p}_v^{\text{res}}$ (non-leaf). If a node is rejected, we proceed to the next node in traversal order until an acceptance or all nodes are rejected. Upon acceptance at a leaf, we sample the next token from $\mathcal{M}_b$; upon acceptance at a non-leaf node, we sample from the residual distribution.

Algorithm 1 UniVer

Require: Draft tree T with root r, target model $\mathcal{M}_b$, draft model $\mathcal{M}_s$
Ensure: Accepted sequence $\mathbf{v}$ and next token y

// Allocation Phase
Initialize $\tilde{p}_r \leftarrow 1$
for each layer $\ell = 0$ to $D - 1$ **do**
  for each node v at layer ℓ (in parallel) **do**
    Compute marginal probabilities $p_v(u_i)$ for all $u_i \in \mathcal{C}(v)$ via Eqs. 4–6
    Compute rejection probability $p_v(\neg v)$ (Eq. 7)
    // Conditional normalization for traversal order
    $m \leftarrow |\mathcal{C}(v)|$
    for $j = 1$ to m **do**
      $\tilde{p}_{u_j} \leftarrow \frac{p_v(u_j)}{1 - \sum_{i=1}^{j-1} p_v(u_i)}$ ▷ Conditional prob.
    end for
    $\tilde{p}_v^{\text{res}} \leftarrow 1 - \frac{p_v(\neg v)}{1 - \sum_{i=1}^{m} p_v(u_i)}$ ▷ Fallback prob.
  end for
end for
// Decision Phase
Generate $\boldsymbol{\eta} \sim U(0, 1)^{|T|}$
for each node v in post-order traversal sequence **do**
  if v is leaf and $\eta_v < \tilde{p}_v$ **then**
    Sample $y \sim \mathcal{M}_b(\cdot \mid v)$ and **return** (v, y)
  end if
  if v is non-leaf and $\eta_v < \tilde{p}_v^{\text{res}}$ **then**
    Sample $y \in \Sigma \setminus \mathcal{C}(v)$ from normalized residual distribution $\frac{p_v(y)}{\sum_{u \notin \mathcal{C}(v)} p_v(u)}$ and **return** (v, y)
  end if
end for

3.2 UniVer Algorithm

We now introduce the technical details of the UniVer pipeline (Algorithm 1), including its sampling strategy and computation of the core acceptance probability.

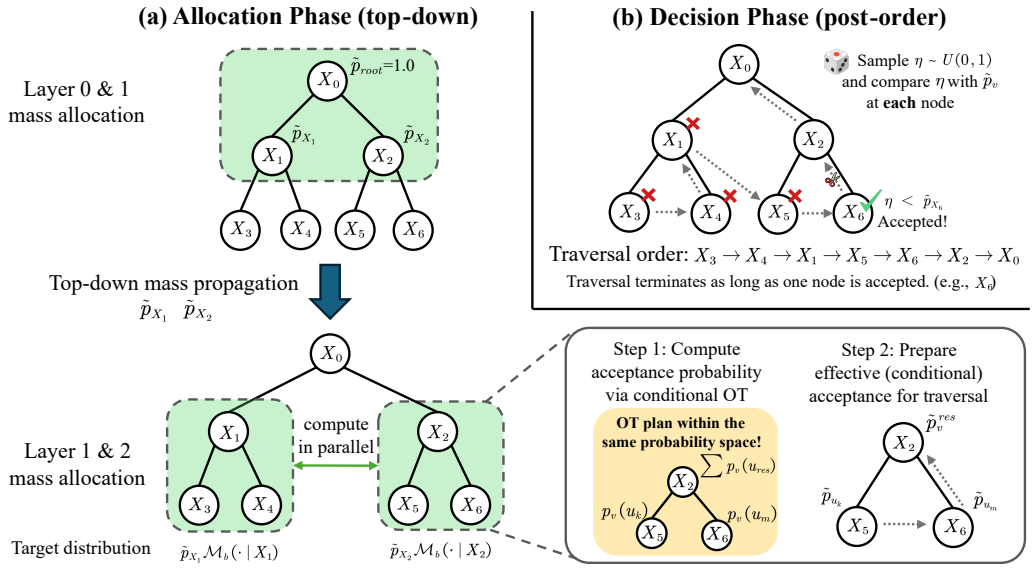


Figure 2: Two-stage verification framework of **UniVer**. **(a) Allocation Phase**: Acceptance probabilities are propagated in a top-down and layer-by-layer manner. For each branch within a layer, marginal probabilities are computed in parallel using conditional optimal transport, then normalized into conditional probabilities used for the next layer's acceptance probability computation. **(b) Decision Phase**: A single-pass post-order traversal is executed. For each node X_i, a random variable η is compared against $\tilde{p}_{X_i}$ (for leaves) or $\tilde{p}^{\text{res}}_{X_i}$ (for non-leaves); the process terminates upon acceptance.

UniVer Acceptance Probability Computation. For any parent node v with effective conditional acceptance probability $\tilde{p}_v$ (precomputed recursively), its children nodes $\mathcal{C}(v) = \{u_1, \ldots, u_m\}$ are selected by Greedy sampling strategy (i.e., Top_{m-1} plus $u_m \sim \mathcal{M}_s^{\neg}$).

For clarity, define the normalization factor:

$$Z_v = 1 - \tilde{p}_v + \sum_{x \in \Sigma} \left[\tilde{p}_v \mathcal{M}_b(x \mid v) - \mathcal{M}_s^{\neg}(x \mid v)\right]_+. \tag{3}$$

The marginal acceptance probabilities are given by:

$$p_v(u_m) = \min\left\{1, \frac{\tilde{p}_v \mathcal{M}_b(u_m \mid v)}{\mathcal{M}_s^{\neg}(u_m \mid v)}\right\} \tag{4}$$

$$p_v(u_k) = \left[\tilde{p}_v \mathcal{M}_b(u_k \mid v) - \mathcal{M}_s^{\neg}(u_k \mid v)\right]_+ \times \frac{1 - p_v(u_m)}{Z_v}, \quad \forall 1 \le k \le m-1, \tag{5}$$

$$p_v(u_{\text{res}}) = \left[\tilde{p}_v \mathcal{M}_b(u_{\text{res}} \mid v) - \mathcal{M}_s^{\neg}(u_{\text{res}} \mid v)\right]_+ \times \frac{1 - p_v(u_m)}{Z_v}, \quad \forall u_{\text{res}} \in \Sigma \setminus \mathcal{C}(v), \tag{6}$$

Then, the remain probability of rejecting v is

$$p_v(\neg v) = 1 - \sum_{u \in \Sigma} p_v(u) = \frac{(1 - p_v(u_m))(1 - \tilde{p}_v)}{Z_v}. \tag{7}$$

ensuring proper probability allocation.

Intuitively, Eqs. (3)–(7) first settle the sampled child u_m as in standard speculative sampling, and then redistribute the probability of rejecting u_m to the other outcomes in proportion to the target mass that u_m cannot cover, *i.e.,* $[\tilde{p}_v \mathcal{M}_b(x) - \mathcal{M}_s^{\neg}(x)]_+$, with Z_v as the normalizer. For a non-root node, the unallotted mass $1 - \tilde{p}_v$ is treated as a virtual outcome $\neg v$ (rejecting v as a whole), so that accepting a child, falling back to v, and rejecting v all lie in a single probability space; the root is the special case $\tilde{p}_r = 1$. Appendix A explains this logic step by step on a two-layer example.

4 Theoretical Guarantees

This section establishes the theoretical foundations of UniVer, proving three key properties:

- **Losslessness.** The algorithm preserves the target model distribution $\mathcal{M}_b$.
- **Conditional Optimality.** UniVer achieves the optimal acceptance rate for each single-layer subtree of T, given a prefix acceptance probability and its sampling strategy.
- **Superiority over Greedy.** The expected acceptance length of our UniVer is never worse than that of the Greedy method [8].

Preliminaries. We first formalize some key notations used in our analysis. Let p_{draft} be the probability distribution of generating a specific draft tree T under the draft model $\mathcal{M}_s$, satisfying $\sum_T p_{\text{draft}}(T) = 1$. For a verification algorithm $\mathscr{A}$, we write $\mathscr{A} = o'$ to indicate that the algorithm outputs exactly the sequence o'. Note that o' except the last one form a valid root-to-node path in T, while its last element is a sample token y. Each *prefix acceptance probability* $\tilde{p}_v \in [0, 1]$ has been evaluated recursively by Line 10 of Algorithm 1.

Detailed proofs of the following theoretical results are referred to Appendix D.

Definition 1 (Locally Lossless Tree)**.** *Let T be a draft tree equipped with acceptance probabilities $\{\tilde{p}_v\}_{v\in T}$ and $\{p_v(u)\}_{u\in\mathcal{C}(v)}$ generated by the* Allocation Phase*. If for any non-leaf node $v \in T$, the following holds: given the prefix v and $\tilde{p}_v$, the generation of children $\mathcal{C}(v)$ is **conditionally independent** of the part of T outside the subtree rooted at v, and*

$$\mathbb{E}_{\mathcal{C}(v)}[p_v(t) \mid \tilde{p}_v] = \tilde{p}_v \mathcal{M}_b(t \mid v), \quad \forall t \in \Sigma, \tag{8}$$

*then T is called a **locally lossless tree**.*

Definition 2 (Lossless Verification)**.** *A verification algorithm $\mathscr{A}(T, \mathcal{M}_s, \mathcal{M}_b)$ is **lossless** if for any sequence $o = (o_0, o_1, \ldots, o_L) \in \Sigma^{L+1}$ with $o_0 = r$ being the root, the following equation holds:*

$$\sum_T p_{\text{draft}}(T) \sum_{o'\subseteq o} \frac{\mathcal{M}_b(o)\Pr[\mathscr{A} = o' \mid T]}{\mathcal{M}_b(o')} = \mathcal{M}_b(o), \tag{9}$$

where the inner summation ranges over all prefix sequences $o' \subseteq o$ (including $o' = o$ itself), and $\frac{\mathcal{M}_b(o)}{\mathcal{M}_b(o')}$ is the conditional probability of generating the remaining tokens $\{o \setminus o'\}$ by $\mathcal{M}_b(\cdot \mid o')$.

Theorem 3 (Local losslessness of UniVer)**.** *The draft tree equipped with the acceptance probabilities generated in* Allocation Phase *of UniVer verification is a locally lossless tree.*

Theorem 4 (General Losslessness)**.** *For any two-stage verification algorithm $\mathscr{A}(T, \mathcal{M}_s, \mathcal{M}_b)$ whose Decision Phase follows Algorithm 1, if the draft tree generated in its Allocation Phase is **locally lossless**, then the verification $\mathscr{A}$ is **lossless**.*

Corollary 1 (Losslessness of UniVer)**.** *From Theorems 3 and 4, UniVer produces tokens are consistent with the exact target distribution $\mathcal{M}_b$, i.e., UniVer is **strictly lossless**.*

Remark. Theorem 4 does not rely on the specific allocation rule in Eqs. (3)–(7): any Allocation Phase that yields a locally lossless tree gives a lossless two-stage verifier. Therefore, other horizontal sampling and verification schemes can also be incorporated into UniVer by replacing these formulas accordingly. For example, using RRSw with sampling without replacement as the per-node primitive recovers Traversal Verification [24]. In contrast, the closed-form allocation, the conditional optimality in Theorem 5, and the superiority over Greedy in Theorem 6 below are specific to Greedy sampling.

Theorem 5 (Conditional Optimality)**.** *Let T be any locally lossless tree satisfying Definition 1 and let $v \in T$ be a non-leaf node with effective acceptance probability $\tilde{p}_v$. Then UniVer achieves the maximum conditional acceptance rate for the children $\mathcal{C}(v) = \{u_1, \ldots, u_m\}$ (over the randomness of $u_m \sim \mathcal{M}_s^{\neg}(\cdot \mid v)$), which equals:*

$$\begin{aligned} &\alpha^*_{\text{UniVer}}(\tilde{p}_v \mathcal{M}_b(\cdot \mid v), \mathcal{M}_s(\cdot \mid v)) \\ =& \sum_{u\in \text{Top}_{m-1}(\mathcal{M}_s(\cdot|v))} \tilde{p}_v \mathcal{M}_b(u \mid v) + \sum_{u\in\Sigma} \min\{\tilde{p}_v \mathcal{M}_b(u \mid v), \mathcal{M}_s^{\neg}(u \mid v)\}. \end{aligned} \tag{10}$$

Theorem 6. *For any draft tree $T \sim p_{\text{draft}}$, let $N_{\text{UniVer}}(T)$ and $N_{\text{Greedy}}(T)$ denote the acceptance lengths of UniVer and the Greedy method [8], respectively. Then UniVer exhibits the following superiority:*

$$\mathbb{E}_{T\sim p_{\text{draft}}}[N_{\text{UniVer}}(T)] \geq \mathbb{E}_{T\sim p_{\text{draft}}}[N_{\text{Greedy}}(T)]. \tag{11}$$

Table 1: Acceptance length (τ) and throughput on Vicuna-7B-v1.3. Autoregressive throughput baseline: 35.9 tokens/s. Best results are highlighted in **bold**.

Method		MT	Trans.	Summ.	QA	Math	RAG	Avg. τ (Δ)	Avg. TPS
RRSw-based	RRSw	3.42±0.03	2.73±0.03	2.97±0.02	2.72±0.01	3.47±0.06	3.05±0.01	3.06±0.01 (0.0%)	67.5±0.2
	Traversal	3.49±0.03	2.75±0.04	3.03±0.02	2.79±0.03	3.55±0.06	3.15±0.06	3.13±0.02 (2.3%↑)	68.9±0.5
OT-based	Greedy	3.56±0.04	2.80±0.03	3.12±0.03	2.87±0.05	3.66±0.06	3.24±0.03	3.21±0.02 (4.9%↑)	70.4±0.6
	UniVer	**3.68±0.02**	**2.83±0.04**	**3.21±0.04**	**2.94±0.02**	**3.76±0.02**	**3.32±0.05**	**3.29±0.01** (7.5%↑)	**72.2±0.5**

Table 2: Acceptance length (τ) and throughput on Llama3.1-8B-Instruct. Autoregressive throughput baseline: 34.5 tokens/s. Best results are highlighted in **bold**.

Method		MT	Trans.	Summ.	QA	Math	RAG	Avg. τ (Δ)	Avg. TPS
RRSw-based	RRSw	2.96±0.03	2.31±0.03	2.56±0.01	2.33±0.01	3.31±0.06	2.86±0.04	2.72±0.02 (0.0%)	48.5±0.2
	Traversal	3.05±0.02	2.34±0.06	2.60±0.01	2.40±0.05	3.37±0.05	2.94±0.04	2.78±0.02 (2.2%↑)	49.5±0.4
OT-based	Greedy	3.09±0.01	2.54±0.04	2.70±0.02	2.46±0.02	3.45±0.03	3.04±0.06	2.88±0.02 (5.9%↑)	50.6±0.4
	UniVer	**3.19±0.02**	**2.57±0.06**	**2.77±0.01**	**2.57±0.04**	**3.51±0.03**	**3.11±0.04**	**2.95±0.01** (8.5%↑)	**52.2±0.1**

5 Experiments

5.1 Experimental Setup

Datasets. We conduct the experiments on Spec-Bench [26], following the setup of Traversal Verification [24]. Spec-Bench encompasses six distinct domains, each with 80 representative samples: multi-turn conversation on MT-Bench [29], translation on WMT14 DE-EN [1], summarization on CNN/Daily Mail [18], question answering on Natural Questions [12], mathematical reasoning on GSM8K [5], and retrieval-augmented generation on DPR [10].

Models. We mainly conduct experiments on the Vicuna [29] model family, using EAGLE [15] as the draft model. This pairing is widely adopted in previous research [19, 24, 8]. We also include experiments on other models such as Llama3.1-8B-Instruct [6] to validate its generalization. Due to space limits, we do not include all results in the main text. Please refer to Appendix B for comprehensive experiments with different drafter structures and model sizes.

Implementation and Comparisons. We implement UniVer and baseline methods based on the EAGLE [15] open-source repository. In UniVer's allocation phase, nodes at the same tree depth are processed in parallel using custom Triton kernels. All experiments are conducted on NVIDIA RTX A6000 48G GPUs, with each configuration running on three different random seeds. We report the *acceptance length* τ (average tokens generated per cycle) by mean$\pm$std as the primary metric, which reflects the theoretical efficiency of the verification algorithm. We also measure the practical throughput (Tokens Per Second, TPS) for a comprehensive comparison. We compare UniVer against existing verification methods spanning two paradigms:

1. **RRSw-based:** (1) ***Vanilla RRSw***: Standard recursive rejection sampling without replacement applied independently at each node, as implemented in EAGLE [15]; (2) ***Traversal***: The Traversal Verification algorithm [24] which extends Block Verification to trees using RRSw for horizontal draft selection.
2. **OT-based:** (1) ***Greedy method*** [8]: Single-step OT with Greedy draft selection; (2) ***UniVer*** (Ours): Our proposed unified framework that extends OT-based verification to tree structures.

5.2 Overall Performance

Table 1 presents the results on Vicuna-7B-v1.3 using a balanced binary tree of depth 5 (32 leaf nodes) under temperature 1.0. UniVer achieves a τ of 3.29, representing a 7.5% improvement over the Vanilla RRSw baseline. Notably, Traversal Verification, which extends Block Verification to trees using RRSw, only marginally outperforms Vanilla RRSw (2.3%), suggesting that bottom-up traversal

alone provides limited benefit when horizontal draft selection remains suboptimal. In contrast, UniVer achieves a 7.5% higher τ over the vanilla baseline and a 2.5% gain over Greedy method [8].

Figure 3 demonstrates the per-depth acceptance rates on MT-bench. At depth 0, Greedy and UniVer both achieve 82.6%, outperforming RRSw-based methods through coordinated horizontal selection. All methods exhibit a characteristic drop at depth 1, consistent with the known degradation of EAGLE draft model alignment beyond the first token [28, 25]. As depth increases, RRSw degrades steadily while Traversal stabilizes around 75% via vertical probability recharging but plateaus. Greedy enjoys strong early-layer acceptance, but shares similar decay patterns like RRSw. UniVer maintains an acceptance rate of at least 77% at every depth, confirming that it successfully propagates the probability mass across depths and benefits from horizontal-vertical joint optimization.

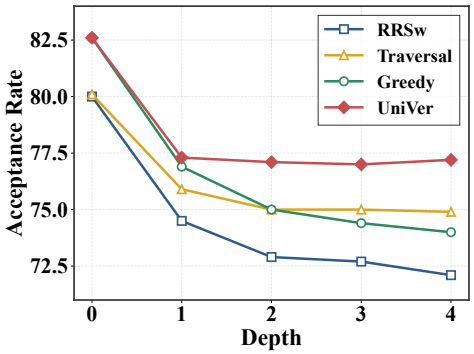


Figure 3: Per-depth acceptance rates on Vicuna-7B-v1.3 (binary tree, depth 5).

In terms of wall-clock efficiency, since UniVer only uses one sampled token with other tokens selected deterministically, the acceptance probabilities of all candidates admit closed-form expressions without using OT solvers. The overall latency of UniVer is therefore comparable to that of standard RRSw, and the acceptance gains directly translate into throughput improvements. To contextualize these numbers: for Vicuna-7B-v1.3, while UniVer improves acceptance length by 7.5%, the end-to-end throughput gain is approximately 7%.

Similar trends are observed on Llama3.1-8B-Instruct (Table 2), confirming its robustness across different architectures. For full experimenetal results, please refer to Appendix B.

5.3 Scaling with Tree Size

As shown in Figure 4, the advantage of UniVer becomes more pronounced as the tree size and depth increase. Compared to single-step OT (Greedy method), UniVer achieves larger gains when the tree gets deeper. This trend arises because UniVer's conditional OT framework effectively propagates probability mass through hierarchical dependencies, whereas Greedy applies independent single-step optimization at each node, as the local truncation of probabilities compounds across hierarchical levels. Similarly, as depth grows, Traversal demonstrates increasing advantages over Vanilla RRSw, validating that sequence-level verification benefits from deeper trees. However, UniVer consistently outperforms Traversal across all depths, as conditional OT achieves superior horizontal draft selection compared to RRSw's local pairwise comparisons.

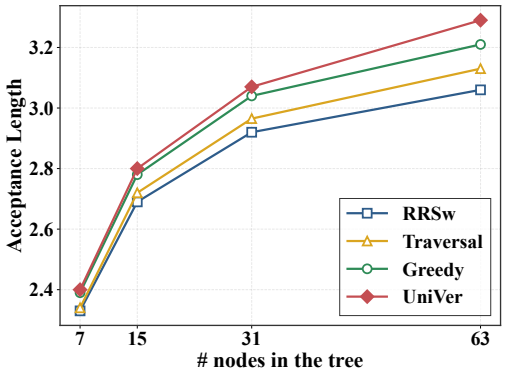


Figure 4: Acceptance length as a function of binary tree size (number of nodes) on Vicuna-7B-v1.3. Tree depth ranges from 2 to 5, corresponding to 7, 15, 31, and 63 nodes respectively. The temperature is set to 1.0.

5.4 Effect of Tree Topology and Temperature

Table 3 examines how UniVer performs across varying tree architectures and sampling temperatures.

Impact of Temperature. As the temperature decreases, the output distributions become more concentrated, causing the performance gap between verification methods to narrow. At temperature 0.3, Traversal, Greedy, and UniVer achieve nearly identical acceptance lengths (*e.g.,* 3.39 for the binary tree with depth = 5 configuration), converging toward the same outcome as observed at temperature 0 (3.40), where the deterministic nature renders these verification methods equivalent.

Table 3: Impact of temperature and tree topology on acceptance length (τ) using Vicuna-7B-v1.3. Results show mean$_{\pm\text{std}}$. Δ indicates relative improvement over Vanilla RRSw.

Method		**Drafts=2 Steps=5**		**Drafts=3 Steps=3**		**EAGLE Tree**	
		Avg. τ	Δ	Avg. τ	Δ	Avg. τ	Δ
Temperature = 0.3							
RRSw-based	Vanilla RRSw	3.34$_{\pm 0.01}$	0.0%	3.04$_{\pm 0.01}$	0.0%	3.33$_{\pm 0.01}$	0.0%
	Traversal	**3.39$_{\pm 0.01}$**	1.5%	3.08$_{\pm 0.01}$	1.3%	**3.38$_{\pm 0.01}$**	1.5%
OT-based	Greedy	**3.39$_{\pm 0.01}$**	1.5%	3.08$_{\pm 0.01}$	1.3%	**3.38$_{\pm 0.01}$**	1.5%
	UniVer	**3.39$_{\pm 0.01}$**	1.5%	**3.09$_{\pm 0.01}$**	1.6%	**3.38$_{\pm 0.01}$**	1.5%
Temperature = 0.6							
RRSw-based	Vanilla RRSw	3.29$_{\pm 0.01}$	0.0%	3.02$_{\pm 0.01}$	0.0%	3.29$_{\pm 0.01}$	0.0%
	Traversal	3.32$_{\pm 0.01}$	0.9%	3.05$_{\pm 0.01}$	1.0%	3.33$_{\pm 0.01}$	1.2%
OT-based	Greedy	3.34$_{\pm 0.02}$	1.5%	3.07$_{\pm 0.01}$	1.7%	3.34$_{\pm 0.01}$	1.5%
	UniVer	**3.37$_{\pm 0.01}$**	2.4%	**3.09$_{\pm 0.01}$**	2.3%	**3.35$_{\pm 0.01}$**	1.8%
Temperature = 1.0							
RRSw-based	Vanilla RRSw	3.06$_{\pm 0.01}$	0.0%	2.89$_{\pm 0.01}$	0.0%	3.11$_{\pm 0.01}$	0.0%
	Traversal	3.13$_{\pm 0.02}$	2.3%	2.91$_{\pm 0.01}$	0.7%	3.16$_{\pm 0.01}$	1.6%
OT-based	Greedy	3.21$_{\pm 0.02}$	4.9%	3.00$_{\pm 0.01}$	3.8%	3.21$_{\pm 0.01}$	3.2%
	UniVer	**3.29$_{\pm 0.01}$**	7.5%	**3.03$_{\pm 0.01}$**	4.8%	**3.24$_{\pm 0.02}$**	4.2%
Reference (Temperature = 0)		3.40	-	3.09	-	3.40	-

Conversely, at higher temperatures, where distributions are more dispersed, UniVer demonstrates more pronounced advantages over existing methods.

Impact of Tree Topology. UniVer maintains consistent improvements across diverse tree structures, including balanced trees (2 drafts $\times$ 5 steps and 3 drafts $\times$ 3 steps) and the imbalanced EAGLE tree. While the absolute gain varies with topology (e.g., 7.5% on the balanced 2x5 tree versus 4.2% on the EAGLE tree at temperature 1.0), UniVer consistently achieves the highest acceptance lengths. This variation arises because the optimal transport formulation allocates probability mass globally across siblings, whereas the relative benefit of such global coordination depends on the specific branching structure and depth distribution of the tree.

6 Related Work

Speculative Decoding. Introduced by Leviathan et al. [13] and Chen et al. [3], speculative decoding accelerates LLM inference via a draft-then-verify paradigm. Research in this area primarily focuses on two directions: (1) improving draft quality and tree structures, including architectural innovations such as EAGLE series [15, 16], Medusa [2] and tree topology design, like EAGLE-2 [14] and Sequoia [4]; (2) designing verification algorithms, which is most relevant to this work.

Multi-Step Verification. Methods in this direction extend speculative decoding beyond single-step generation to handle sequential dependencies. For single-chain decoding, Block Verification [20] and ASpS [7] identify the sub-optimality of local (per-step) verification and propose sequence-level alternatives that maximize expected acceptance length along the chain. Weng et al. [24] further extend this to tree structures through Traversal Verification, which processes nodes bottom-up.

Multi-Draft Verification. Complementary to multi-step methods, this direction focuses on optimizing the acceptance rate for multiple candidates generated at a single step. SpecTr [21] formulates verification as an optimal transport problem and has been further improved by subsequent works [11, 22]. SpecInfer [17] employs Recursive Rejection Sampling (RRS), later refined to RRSw [9, 27, 4] to prevent redundant sampling of identical tokens. Beyond standard sampling strategies, SpecHub [19] and Greedy method [8] propose hybrid approaches that deterministically select high-probability candidates with stochastic sampled tokens, with the latter achieving the theoretical optimal acceptance rate under its drafting strategy within single-step.

7 Conclusion

In this work, we addressed the fragmentation in speculative decoding verification, where existing methods treat multi-draft and multi-step optimizations in isolation. We revealed that longitudinal dependencies can be formulated as dynamic scaling factors via prefix acceptance probabilities and framed tree-based verification as a conditional OT problem.

Building on this theoretical foundation, we proposed UniVer, an algorithm that composes local optimal transport plans under prefix constraints to jointly optimize across tree levels. We proved that UniVer achieves the optimal acceptance rate within this conditional framework while maintaining strict losslessness. Moreover, losslessness holds for any allocation that yields a locally lossless tree, so UniVer offers a general way to extend single-step horizontal verifiers to multi-step tree verification. Empirical results demonstrate that UniVer significantly improves acceptance length over existing methods, validating the efficacy of our unified verification strategy.

Limitations and Future Work. Distinct verification methods only exhibit performance differences when the sampling temperature is above zero. Therefore, UniVer possesses no extra performance gain under or near temperature = 0. Beyond this universal boundary, the realized acceptance length is intrinsically linked to the interplay between tree topology and sampling strategy. Although the UniVer framework admits other horizontal verifiers, its closed-form allocation and conditional optimality are derived under the specific mixed (Greedy) sampling strategy, while co-designing the sampling strategy, verification mechanism, and the tree structure to maximize longitudinal mass propagation remains a promising direction for future research. Moreover, although UniVer is optimal within the conditional framework, it does not always attain the theoretical upper bound given by the global joint-OT formulation (Appendix C). A better verifier therefore exists in principle, but how to construct it and whether its practical overhead is affordable remain open problems.

Acknowledgments and Disclosure of Funding

This work was supported by the University of Tokyo and Lenovo AI Technology Center. The authors declare no competing interests.

References

[1] Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*.

[2] 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 International Conference on Machine Learning*.

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

[4] Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. 2024. Sequoia: Scalable and robust speculative decoding. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[7] Zhengmian Hu and Heng Huang. 2024. Accelerated speculative sampling based on tree monte carlo. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

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

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

[10] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*.

[11] Ashish J Khisti, MohammadReza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. 2025. Multi-draft speculative sampling: Canonical decomposition and theoretical limits. In *The Thirteenth International Conference on Learning Representations*.

[12] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*.

[13] Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *Proceedings of the International Conference on Machine Learning*.

[14] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. EAGLE-2: faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*.

[15] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. EAGLE: speculative sampling requires rethinking feature uncertainty. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

[16] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[17] Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 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, ASPLOS 2024, La Jolla, CA, USA, 27 April 2024- 1 May 2024*.

[18] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*.

[19] Ryan Sun, Tianyi Zhou, Xun Chen, and Lichao Sun. 2024. SpecHub: Provable acceleration to multi-draft speculative decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

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

[21] Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix X. Yu. 2023. Spectr: Fast speculative decoding via optimal transport. In *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*.

[22] Rahul Krishna Thomas and Arka Pal. 2025. Global resolution: Optimal multi-draft speculative sampling via convex minimization. *arXiv preprint arXiv:2511.15898*.

[23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models.

[24] Yepeng Weng, Qiao Hu, Xujie Chen, Li Liu, Dianwen Mei, Huishi Qiu, Jiang Tian, and Zhongchao Shi. 2025. Traversal verification for speculative tree decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[25] Yepeng Weng, Dianwen Mei, Huishi Qiu, Xujie Chen, Li Liu, Jiang Tian, and Zhongchao Shi. 2025. CORAL: Learning consistent representations across multi-step training with lighter speculative drafter. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5580–5593, Vienna, Austria.

[26] Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*.

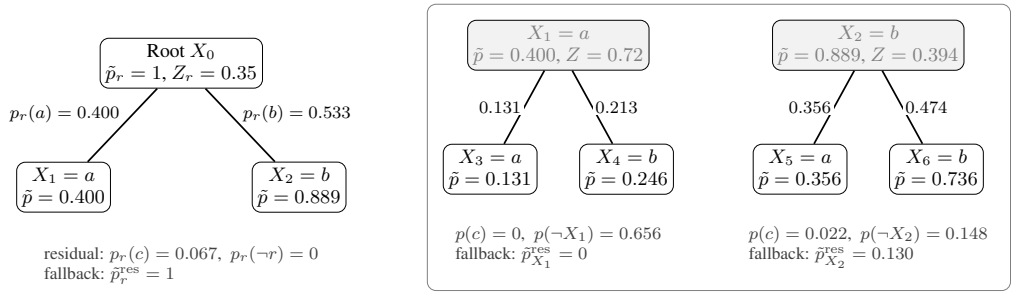
[27] Sen Yang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2024. Multi-candidate speculative decoding. *arXiv preprint arXiv:2401.06706*.

[28] Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. 2025. Learning harmonized representations for speculative sampling. In *International Conference on Learning Representations*.

[29] 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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *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*.

A A Worked Two-Layer Example of UniVer

To make the two phases of UniVer concrete, we walk through a small two-layer tree, using the same toy distributions as the illustrative example of Traversal Verification [24]. Let $\Sigma = \{a, b, c\}$ and assume for simplicity that the distributions do not depend on the prefix: $\mathcal{M}_b = (0.3, 0.4, 0.3)$ and $\mathcal{M}_s = (0.6, 0.3, 0.1)$ over (a, b, c). Each node expands $m = 2$ children by Greedy sampling: the top-1 token a is selected deterministically, and the second child is sampled from the residual draft distribution $\mathcal{M}_s^{\neg} = (0, 0.75, 0.25)$. We consider the tree in Figure 5, where the sampled child is b at every node. All numbers are rounded to three decimals, while intermediate computations use unrounded values.



(a) Layer 0: allocation at the root

(b) Layer 1: allocation at X_1 and X_2 (in parallel)

Figure 5: Allocation Phase on a two-layer draft tree with $\mathcal{M}_b = (0.3, 0.4, 0.3)$ and $\mathcal{M}_s = (0.6, 0.3, 0.1)$ over (a, b, c). Edge labels give the marginal acceptance probabilities $p_v(u)$, and each child node shows its conditional acceptance probability $\tilde{p}$ (Line 10 of Algorithm 1). (a) The root allocates the full target mass ($\tilde{p}_r = 1$). (b) Each node in Layer 1 allocates its scaled target $\tilde{p}_v \mathcal{M}_b$ to its own children; nodes already processed in Layer 0 are shown in gray.

Allocation Phase. We explain the logic behind Eqs. (3)–(7) in four steps, using the tree in Figure 5 as a running example. These steps describe how the probabilities are allocated; the actual accept/reject decisions are made later in post-order.

Step 1: Settle the sampled child. Consider first the root (Figure 5a), which is allotted the full target distribution ($\tilde{p}_r = 1$). Once the top-$(m-1)$ children are fixed deterministically, the only randomness

lies in the sampled child $u_m \sim \mathcal{M}_s^{\neg}$. As in standard speculative sampling, it is accepted with probability $p_r(u_m) = \min\{1, \mathcal{M}_b(u_m)/\mathcal{M}_s^{\neg}(u_m)\}$; here, $p_r(b) = \min\{1, 0.4/0.75\} = 0.533$.

Step 2: Redistribute the remaining mass in a unified probability space. For each token x, the target mass that the sampled child cannot cover is $[\mathcal{M}_b(x) - \mathcal{M}_s^{\neg}(x)]_+$. This includes the top-$(m-1)$ siblings, whose entire target mass remains since $\mathcal{M}_s^{\neg}(x) = 0$, as well as the tokens outside the tree; here, it equals $(0.3, 0, 0.05)$ over (a, b, c). The normalization factor

$$Z_r = \sum_x [\mathcal{M}_b(x) - \mathcal{M}_s^{\neg}(x)]_+ = 1 - \sum_x \min\{\mathcal{M}_b(x), \mathcal{M}_s^{\neg}(x)\} = 0.35$$

is the total uncovered mass. It equals the expected probability that the sampled child is rejected, and it does not depend on which token is sampled (Eq. (13)). When u_m is rejected, which happens with probability $1 - p_r(u_m)$, this probability is split among the tokens in proportion to their uncovered mass, *i.e.,* token x receives the share $[\mathcal{M}_b(x) - \mathcal{M}_s^{\neg}(x)]_+/Z_r$ (Eqs. (5)–(6)). Here, a and c take $0.3/0.35$ and $0.05/0.35$ of the rejection probability $1 - 0.533$, giving

$$p_r(a) = \frac{0.3}{0.35} \times (1 - 0.533) = 0.400, \qquad p_r(c) = \frac{0.05}{0.35} \times (1 - 0.533) = 0.067.$$

As a result, accepting the sampled child, accepting a top-$(m-1)$ sibling, and falling back to the parent (*i.e.,* accepting the parent and drawing a residual token such as c) all lie in a single probability space that sums to one. To see why this allocation is lossless, we average over the randomness of the sampled child. Each token x receives mass $\mathcal{M}_s^{\neg}(x) \min\{1, \mathcal{M}_b(x)/\mathcal{M}_s^{\neg}(x)\} = \min\{\mathcal{M}_b(x), \mathcal{M}_s^{\neg}(x)\}$ through direct acceptance. Since the expected rejection probability of the sampled child is exactly Z_r, the expected mass redistributed to x after rejection is exactly $[\mathcal{M}_b(x) - \mathcal{M}_s^{\neg}(x)]_+$. The two parts sum to the target mass,

$$\underbrace{\min\{\mathcal{M}_b(x), \mathcal{M}_s^{\neg}(x)\}}_{\text{direct acceptance}} + \underbrace{[\mathcal{M}_b(x) - \mathcal{M}_s^{\neg}(x)]_+}_{\text{redistribution}} = \mathcal{M}_b(x),$$

so the local allocation preserves the target distribution.

Step 3: Extend to non-root nodes via an augmented target. A non-root node v has a scaled target $\tilde{p}_v \mathcal{M}_b$, whose total mass is $\tilde{p}_v$. We assign the remaining mass $1 - \tilde{p}_v$ to a virtual rejection outcome $\neg v$, forming a normalized augmented target distribution over $\Sigma \cup \{\neg v\}$. This specifies the rejection probability of v in expectation over the sampled child; the rejection probability allocated for a particular sampled child can differ. Since $\mathcal{M}_s^{\neg}(\neg v) = 0$, this virtual outcome can never be covered by the sampled child, just like a top-$(m-1)$ sibling. Applying Steps 1 and 2 to this augmented distribution recovers Eqs. (3)–(7): the normalization factor Z_v gains the additional term $1 - \tilde{p}_v$, and the virtual outcome receives $p_v(\neg v) = (1 - p_v(u_m))(1 - \tilde{p}_v)/Z_v$. The root is simply the special case $\tilde{p}_r = 1$. For instance, X_1 in Figure 5b has $\tilde{p}_{X_1} = 0.4$ and the augmented target $(0.12, 0.16, 0.12;\ 0.6)$ over $(a, b, c;\ \neg X_1)$. Its sampled child b is accepted with probability $\min\{1, 0.16/0.75\} = 0.213$, the uncovered mass is $(0.12, 0, 0;\ 0.6)$ with $Z_{X_1} = 0.72$, and the rejection probability $1 - 0.213$ is split into

$$p_{X_1}(a) = \frac{0.12}{0.72} \times (1 - 0.213) = 0.131, \qquad p_{X_1}(\neg X_1) = \frac{0.6}{0.72} \times (1 - 0.213) = 0.656.$$

Since only 40% of the target mass is allotted to X_1, most of its rejection probability is allocated to $\neg X_1$ (here 0.656 for the sampled child b, versus 0.6 in expectation). In the Decision Phase, this outcome means that the subtree of X_1 is rejected and the traversal continues to the subtree of X_2. In parallel, X_2 with $\tilde{p}_{X_2} = 0.889$ yields $p_{X_2}(b) = 0.474$, $p_{X_2}(a) = 0.356$, $p_{X_2}(c) = 0.022$, and $p_{X_2}(\neg X_2) = 0.148$.

Step 4: Convert marginal into conditional probabilities. The probabilities above are marginal within the probability space of v, whereas the Decision Phase visits the children one by one. Since a child u_j is visited only after all its earlier siblings have been rejected, its marginal probability is converted into the conditional probability $\tilde{p}_{u_j} = p_v(u_j)/(1 - \sum_{i<j} p_v(u_i))$ (Line 10 of Algorithm 1). Likewise, v itself is reconsidered only after all its children are rejected, so its fallback probability $\tilde{p}_v^{\text{res}}$ is the

residual-token mass conditioned on this event (Line 12). This yields the values in Figure 5:

$$\begin{aligned}
&\text{root:} && \tilde{p}_{X_2} = \frac{0.533}{1-0.400} = 0.889, && \tilde{p}_r^{\text{res}} = 1,\\
&X_1: && \tilde{p}_{X_4} = \frac{0.213}{1-0.131} = 0.246, && \tilde{p}_{X_1}^{\text{res}} = 0,\\
&X_2: && \tilde{p}_{X_6} = \frac{0.474}{1-0.356} = 0.736, && \tilde{p}_{X_2}^{\text{res}} = \frac{0.022}{0.022+0.148} = 0.130,
\end{aligned}$$

where $\tilde{p}_{X_1}^{\text{res}} = 0$ because X_1 has no residual-token mass left. The conditional probability of a non-leaf child (*e.g.,* $\tilde{p}_{X_1} = 0.400$ and $\tilde{p}_{X_2} = 0.889$) then serves as its scaling factor in the next layer.

Decision Phase. With the probabilities computed above, we visit the nodes in post-order: $X_3 \to X_4 \to X_1 \to X_5 \to X_6 \to X_2 \to X_0$. At each visited node v, we accept with probability $\tilde{p}_v$ for a leaf or $\tilde{p}_v^{\text{res}}$ for a non-leaf node, stopping at the first acceptance. In this example, X_1 is always skipped because $\tilde{p}_{X_1}^{\text{res}} = 0$. Upon acceptance, we output the corresponding root-to-node prefix and sample one additional token from the target distribution at a leaf, or from the residual distribution at a non-leaf node.

B Additional Experimental Results

We provide additional experimental results on Vicuna-13B-v1.3, Vicuna-33B-v1.3 and Qwen2-7B-Instruct with their corresponding EAGLE draft model in Table 4, Table 5, and Table 6. We also include experiments on Llama2-7B [23] with Llama-68M [17] as the draft model in Table 7.

Table 4: Acceptance length (τ) and throughput on Vicuna-13B-v1.3. Autoregressive throughput baseline: 22.3 tokens/s. Best results are highlighted in **bold**.

Method		MT	Trans.	Summ.	QA	Math	RAG	Avg. τ (Δ)	Avg. TPS
RRSw-based	RRSw	3.57±0.06	2.81±0.03	3.17±0.03	2.77±0.05	3.63±0.08	3.11±0.05	3.17±0.02 (0.0%)	44.3±0.2
	Traversal	3.60±0.05	2.82±0.02	3.19±0.03	2.85±0.05	3.71±0.05	3.15±0.04	3.22±0.01 (1.6%↑)	44.9±0.1
OT-based	Greedy	3.67±0.02	2.92±0.04	3.27±0.01	2.91±0.04	3.79±0.03	3.28±0.04	3.30±0.02 (4.1%↑)	45.7±0.3
	UniVer	**3.74±0.02**	**2.97±0.03**	**3.31±0.03**	**2.97±0.01**	**3.84±0.10**	**3.30±0.03**	**3.35±0.01** (5.7%↑)	**46.5±0.4**

Table 5: Acceptance length (τ) and throughput on Vicuna-33B-v1.3. Autoregressive throughput baseline: 9.2 tokens/s. Best results are highlighted in **bold**.

Method		MT	Trans.	Summ.	QA	Math	RAG	Avg. τ (Δ)	Avg. TPS
RRSw-based	RRSw	3.38±0.04	2.68±0.01	2.97±0.01	2.66±0.02	3.70±0.01	2.94±0.02	3.05±0.02 (0.0%)	20.6±0.1
	Traversal	3.41±0.01	2.72±0.01	3.01±0.01	2.68±0.04	3.73±0.04	2.98±0.03	3.09±0.01 (1.1%↑)	20.8±0.1
OT-based	Greedy	3.50±0.01	2.80±0.02	3.07±0.01	2.78±0.01	3.82±0.03	3.05±0.03	3.17±0.01 (3.8%↑)	21.2±0.1
	UniVer	**3.52±0.01**	**2.83±0.02**	**3.11±0.03**	**2.81±0.06**	**3.86±0.01**	**3.10±0.05**	**3.21±0.01** (4.9%↑)	**21.4±0.1**

Table 6: Acceptance length (τ) and throughput on Qwen2-7B-Instruct. Autoregressive throughput baseline: 37.0 tokens/s. Best results are highlighted in **bold**.

Method		MT	Trans.	Summ.	QA	Math	RAG	Avg. τ (Δ)	Avg. TPS
RRSw-based	RRSw	2.09±0.01	2.57±0.03	1.44±0.01	2.37±0.03	3.09±0.02	1.48±0.02	2.17±0.01 (0.0%)	43.7±0.3
	Traversal	2.14±0.01	2.61±0.03	1.45±0.01	2.43±0.04	3.12±0.03	1.50±0.02	2.21±0.01 (1.8%↑)	44.0±0.3
OT-based	Greedy	2.15±0.02	2.64±0.04	1.49±0.02	2.46±0.05	3.21±0.01	1.53±0.02	2.25±0.02 (3.7%↑)	44.7±0.3
	UniVer	**2.20±0.01**	**2.70±0.04**	**1.50±0.01**	**2.57±0.04**	**3.26±0.02**	**1.56±0.02**	**2.30±0.01** (6.0%↑)	**45.8±0.2**

Table 7: Acceptance length (τ) and throughput on Llama2-7B. Autoregressive throughput baseline: 37.3 tokens/s. Best results are highlighted in **bold**.

Method		MT	Trans.	Summ.	QA	Math	RAG	Avg. τ (Δ)	Avg. TPS
RRSw-based	RRSw	2.47±0.03	2.41±0.04	2.15±0.03	2.60±0.02	2.45±0.06	2.52±0.11	2.43±0.04 (0.0%)	57.9±1.3
	Traversal	2.55±0.04	2.49±0.09	2.22±0.04	2.65±0.06	2.58±0.06	2.69±0.02	2.53±0.03 (4.1%↑)	59.6±1.1
OT-based	Greedy	2.48±0.04	2.52±0.07	2.27±0.03	2.68±0.04	2.53±0.06	2.71±0.04	2.53±0.03 (4.1%↑)	59.1±1.0
	UniVer	**2.73±0.01**	**2.59±0.09**	**2.31±0.04**	**2.84±0.04**	**2.66±0.04**	**2.77±0.07**	**2.65±0.03** (9.1%↑)	**62.5±0.4**

C Gap to the Global Joint-OT Optimum

Theorem 5 establishes that UniVer is optimal under the conditional framework, yet its top-down decomposition need not attain the global joint-OT optimum over the whole tree. Here we quantify this gap. Using the notation of Section 4, under complete information the tree verification problem can be written as the following linear program (LP):

$$\begin{aligned} \max_{S_{o,T}\geq 0} \ & \sum_{o,T} (|o|-2)\, S_{o,T} \\ \text{s.t.} \ & \sum_{o'\subseteq o} \frac{\mathcal{M}_b(o)}{\mathcal{M}_b(o')} \sum_{T} S_{o',T} \leq \mathcal{M}_b(o) \ \ (\forall o), \qquad \sum_{o} S_{o,T} \leq p_{\text{draft}}(T) \ \ (\forall T), \end{aligned} \tag{12}$$

where $S_{o,T}$ denotes the probability that tree T is sampled and the verifier outputs o, and $|o|-2$ is the number of accepted draft tokens (excluding the root $o_0 = r$ and the final token sampled from the target model, consistent with $N_{\mathscr{A}}(T)$ in Appendix D). For single-step drafts, Eq. (12) degenerates to the formulation of Hu et al. [8]. Every lossless verifier (including UniVer) induces a feasible solution of Eq. (12), so its optimum upper-bounds the best achievable acceptance length, while UniVer provides a constructive lower bound:

$$\text{UniVer} \ \leq \ \text{Global joint-OT optimum} \ \leq \ \text{LP value}.$$

Solving Eq. (12) is intractable in general, since its state space grows with the tree depth and the number of nodes. In a minimal two-layer example, however, the bound can be computed in closed form.

Setup. Let $\Sigma = \{a, b, c\}$ with root context r and distributions over (a, b, c):

- Target: $\mathcal{M}_b(\cdot \mid r) = (0.6, 0.2, 0.2)$, $\mathcal{M}_b(\cdot \mid r, b) = (0.05, 0.05, 0.9)$, $\mathcal{M}_b(\cdot \mid r, c) = (0.5, 0.25, 0.25)$;
- Draft: $\mathcal{M}_s(\cdot \mid r) = (0.7, 0.15, 0.15)$, $\mathcal{M}_s(\cdot \mid r, b) = \mathcal{M}_s(\cdot \mid r, c) = (0.5, 0.25, 0.25)$.

The tree is built by Greedy sampling with $m = 2$. At depth 1, the root expands a (top-1, leaf) and $s_1 \sim \text{Unif}\{b, c\}$ (sampled, non-leaf); at depth 2, s_1 expands a (top-1, leaf) and $s_2 \sim \text{Unif}\{b, c\}$ (sampled, leaf). This yields four equally likely trees $T_1, \ldots, T_4$ with $(s_1, s_2) \in \{(b, b), (b, c), (c, b), (c, c)\}$, each with probability $1/4$.

Analysis. The chain (b, c) exists only in T_2 and carries target mass $\mathcal{M}_b(bc) = 0.2 \times 0.9 = 0.18$. At the root, Eqs. (3)–(7) give $p_r(a) = 0.6$ and $p_r(b) = 0.4$, so the subtree of b is entered with probability 0.4 and conditional probability $\tilde{p}_b = 1$. The accepted mass of UniVer on this chain is therefore

$$A_{\text{UniVer}}(bc) = \Pr[T_2] \times p_r(b) \times \min\Big\{1, \tilde{p}_b \, \frac{\mathcal{M}_b(c \mid r, b)}{\mathcal{M}_s^{\neg}(c \mid r, b)}\Big\} = \frac{1}{4} \times 0.4 \times 1 = 0.10,$$

whereas any lossless verifier satisfies $A(bc) \leq \min\{\mathcal{M}_b(bc), \Pr[T_2]\} = 0.18$. This bound is attained by accepting the chain with conditional probability 0.72 within T_2. On all other chains, UniVer already matches the upper bound (Table 8).

UniVer thus attains $1.320/1.400 = 94.3\%$ of the globally optimal expected accepted length. In this example the LP upper bound is tight: it can be attained by propagating the untruncated scaling factor $0.9/0.5 = 1.8$ into the next allocation step rather than clipping it at 1. This factor should be

Table 8: Marginal accepted mass of each depth-2 chain and the expected number of accepted draft tokens $\mathbb{E}[N]$ in the minimal two-layer example. Depth-1 tokens are always accepted, contributing 1 to $\mathbb{E}[N]$.

Chain (s_1, y)	(b,a)	(b,b)	(b,c)	(c,a)	(c,b)	(c,c)	$\mathbb{E}[N]$
Target mass $\mathcal{M}_b(s_1 y)$	0.01	0.01	0.18	0.10	0.05	0.05	–
UniVer $A(s_1 y)$	0.01	0.01	**0.10**	0.10	0.05	0.05	1.320
Upper bound (LP)	0.01	0.01	**0.18**	0.10	0.05	0.05	1.400

interpreted as an auxiliary scaling weight rather than a valid acceptance probability. Such over-unity weights fall outside the current UniVer framework, and extending them to general branching trees while preserving shared-prefix marginals is nontrivial.

D The Proof of Theorems

Vocabulary and Sequences. Let Σ denote a finite vocabulary. For any positive integer L, we use Σ^L denote the set of all sequences (strings) of length L over Σ. We write $a = (a_0, \ldots, a_L) \in \Sigma^{L+1}$ to represent a sequence of length $|a| = L + 1$, and $a_{[:\ell]}$ to represent the prefix $(a_0, \ldots, a_\ell)$ consisting of the first ℓ elements of a.

D.1 Local losslessness of UniVer

Proof of Theorem 3. Considering the draft tree T generated by *Allocation Phase* of UniVer verification (see Section 3.2), we observe that, for any fixed non-leaf node $v \in T$, the randomness of its children set $\mathcal{C}(v) = \{u_1, \ldots, u_m\}$ only depends on the last sampling token $u_m \sim \mathcal{M}_s^{\neg}(\cdot \mid v)$, which means $\Pr(\text{draft } \mathcal{C}(v)) = \mathcal{M}_s^{\neg}(u_m \mid v)$. Throughout this proof, we write $\mathbb{E}_{\mathcal{C}(v)}[\cdot]$ as shorthand for $\mathbb{E}_{\mathcal{C}(v)}[\cdot \mid \tilde{p}_v]$, since $\tilde{p}_v$ is fixed when processing node v and the only remaining randomness is in the sampling of u_m.

Now we begin to prove $\mathbb{E}_{\mathcal{C}(v)}[p_v(t)] = \tilde{p}_v \mathcal{M}_b(t|v)$ holds for all $t \in \Sigma$. First, observe that the normalization factor Z_v in Equations (3) is actually independent of the sampled value u_m. We use $H_v = \{u_1, \ldots, u_{m-1}\}$ represent the top-$(m-1)$ tokens, and then

$$
\begin{aligned}
Z_v &= 1 - \tilde{p}_v + \sum_{x \in \Sigma} [\tilde{p}_v \mathcal{M}_b(x|v) - \mathcal{M}_s^{\neg}(x|v)]_+ \\
&= 1 - \tilde{p}_v + \sum_{x \in H_v} \tilde{p}_v \mathcal{M}_b(x|v) + \sum_{x \notin H_v} [\tilde{p}_v \mathcal{M}_b(x|v) - \mathcal{M}_s^{\neg}(x|v)]_+ \\
&= 1 - \tilde{p}_v \sum_{x \notin H_v} \mathcal{M}_b(x|v) + \sum_{x \notin H_v} [\tilde{p}_v \mathcal{M}_b(x|v) - \mathcal{M}_s^{\neg}(x|v)]_+ \\
&= 1 - \sum_{x \notin H_v} \min\{\tilde{p}_v \mathcal{M}_b(x|v), \mathcal{M}_s^{\neg}(x|v)\}. \qquad (13)
\end{aligned}
$$

For any $t \in \Sigma$, we consider the following two cases.

Case 1: $t \in H_v$ **(deterministic nodes).** From Equation (5) and noting that $\mathcal{M}_s^{\neg}(t \mid v) = 0$ for $t \in H_v$, we have

$$
\mathbb{E}_{\mathcal{C}(v)}[p_v(t)] = \frac{\tilde{p}_v \mathcal{M}_b(t \mid v)}{Z_v} \cdot \mathbb{E}_{u_m}[1 - p_v(u_m)].
$$

By Equation (4), we know

$$
\mathbb{E}_{u_m}[1 - p_v(u_m)] = 1 - \sum_{x \notin H_v} \min\left\{\mathcal{M}_s^{\neg}(x|v), \tilde{p}_v \mathcal{M}_b(x|v)\right\}.
$$

The expectation term equals Z_v (see Equation (13)). Thus $\mathbb{E}_{u_m}[p_v(t)] = \tilde{p}_v \mathcal{M}_b(t \mid v)$.

Case 2: $t \notin H_v$ **(residual vocabulary).** Here t may either be the sampled node u_m or belong to the remaining residual set. Obviously,

$$\mathbb{E}_{u_m}[p_v(t)] = \mathcal{M}_s^{\neg}(t \mid v) \cdot \min\left\{1, \frac{\tilde{p}_v \mathcal{M}_b(t \mid v)}{\mathcal{M}_s^{\neg}(t \mid v)}\right\} + \frac{[\tilde{p}_v \mathcal{M}_b(t \mid v) - \mathcal{M}_s^{\neg}(t \mid v)]_+}{Z_v} \times \sum_{x \notin H_v \cup \{t\}} \mathcal{M}_s^{\neg}(x \mid v) \cdot (1 - p_v(x)).$$

Since

$$\begin{aligned}
&\sum_{x \notin H_v \cup \{t\}} \mathcal{M}_s^{\neg}(x \mid v)(1 - p_v(x)) \\
&= \sum_{x \notin H_v \cup \{t\}} \mathcal{M}_s^{\neg}(x \mid v) - \sum_{x \notin H_v \cup \{t\}} \min\{\mathcal{M}_s^{\neg}(x \mid v), \tilde{p}_v \mathcal{M}_b(x \mid v)\} \\
&= Z_v + \min\{0, \tilde{p}_v \mathcal{M}_b(t \mid v) - \mathcal{M}_s^{\neg}(t \mid v)\}.
\end{aligned}$$

Substituting back, we obtain

$$\begin{aligned}
\mathbb{E}_{u_m}[p_v(t)] &= \min\{\mathcal{M}_s^{\neg}(t \mid v), \tilde{p}_v \mathcal{M}_b(t \mid v)\} + [\tilde{p}_v \mathcal{M}_b(t \mid v) - \mathcal{M}_s^{\neg}(t \mid v)]_+ \\
&= \tilde{p}_v \mathcal{M}_b(t \mid v).
\end{aligned}$$

Therefore, the draft tree generated in UniVer verification is a locally lossless tree. □

D.2 Losslessness of two-stage verification

Proof of Theorem 4. We prove it by induction on the number of parent nodes n in tree T.

Base case ($n = 1$)**:** The tree T consists of only the root node r and its children $\mathcal{C}(r) = \{u_1, \ldots, u_m\}$. The depth of T is 1 (root at depth 0, children at depth 1), so we consider any output sequence $o = (r, w, y) \in \Sigma^3$ where $w, y \in \Sigma$.

In the Decision Phase, the traversal order is $(u_1, \ldots, u_m, r)$. There are two cases for how the algorithm $\mathscr{A}$ can generate o:

Case 1: $w \in \mathcal{C}(r)$. The algorithm $\mathscr{A}$ accepts the child node $w = u_i$ for some $1 \le i \le m$ and then samples $y \sim \mathcal{M}_b(\cdot \mid r, w)$. The probability of accepting node w is:

$$\tilde{p}_{u_i} \prod_{j<i} (1 - \tilde{p}_{u_j}) = p_r(w),$$

where $p_r(w)$ is the marginal acceptance mass computed in the Allocation Phase. So the probability of $\mathscr{A}$ outputs o equals

$$\Pr[\mathscr{A} = o \mid w \in \mathcal{C}(r)] = p_r(w) \mathcal{M}_b(y \mid r, w).$$

Then the contribution to the left-hand side of Equation (9) is

$$\begin{aligned}
&\sum_{o' \sqsubseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T]}{\mathcal{M}_b(o')} \\
&= \Pr[\mathscr{A} = o \mid w \in \mathcal{C}(r)] \\
&= p_r(w) \mathcal{M}_b(y \mid r, w), \quad \text{(when } w \in \mathcal{C}(r))
\end{aligned}$$

Case 2: $w \notin \mathcal{C}(r)$. The algorithm $\mathscr{A}$ rejects all children and accepts the root node r, then samples w from the residual distribution $\mathcal{M}_b(\cdot \mid r)$. The probability of accepting the root (fallback) is:

$$\prod_{j=1}^{m} (1 - \tilde{p}_{u_j}) = 1 - \sum_{u \in \mathcal{C}(r)} p_r(u) = \sum_{u \notin \mathcal{C}(r)} p_r(u).$$

So the probability of $\mathscr{A}$ outputs $o' = (r, w)$ equals

$$\Pr[\mathscr{A} = (r, w) \mid w \notin \mathcal{C}(r)] = \left(\sum_{u \notin \mathcal{C}(r)} p_r(u)\right) \frac{p_r(w)}{\sum_{u \notin \mathcal{C}(r)} p_r(u)} = p_r(w).$$

Then the contribution to the left-hand side of Equation (9) is:

$$\begin{aligned}
&\sum_{o' \subseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T]}{\mathcal{M}_b(o')} \\
&= \mathcal{M}_b(y \mid r, w) \Pr[\mathscr{A} = (r, w) \mid w \notin \mathcal{C}(r)] \\
&= p_r(w) \mathcal{M}_b(y \mid r, w), \quad \text{(when } w \notin \mathcal{C}(r))
\end{aligned}$$

Combining both cases, for any sequence $o = (r, w, y)$, we have:

$$\begin{aligned}
&\sum_T p_{\text{draft}}(T) \sum_{o' \subseteq o} \frac{\mathcal{M}_b(o)}{\mathcal{M}_b(o')} \Pr[\mathscr{A} = o' \mid T] \\
&= \sum_T p_{\text{draft}}(T) \cdot p_r(w) \mathcal{M}_b(y \mid r, w) \\
&= \mathcal{M}_b(y \mid r, w) \mathbb{E}_{\mathcal{C}(r)}[p_r(w)].
\end{aligned}$$

By the locally lossless property (Definition 1), for any specific token $w \in \Sigma$, we have $\mathbb{E}_{\mathcal{C}(r)}[p_r(w)] = \tilde{p}_r \mathcal{M}_b(w \mid r) = \mathcal{M}_b(w \mid r)$. Thus, the base case $n = 1$ holds.

Inductive step. Assume the theorem holds for all locally lossless trees with at most K parent nodes. Consider a locally lossless tree T with $K + 1$ parent nodes. Let v be a deepest parent node in T, which implies all children $\mathcal{C}(v)$ are leaves. Construct the reduced tree $T^{\text{sub}} = T \setminus \mathcal{C}(v)$ by removing all children of v. Then T^{sub} contains exactly K parent nodes and satisfies the induction hypothesis, that is for any sequence $o \in \Sigma^{L+1}$ (with L be the length of the longest chain in the tree T),

$$\mathbb{E}_{T^{\text{sub}}} \left[\sum_{o' \subseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T^{\text{sub}}]}{\mathcal{M}_b(o')} \right] = \mathcal{M}_b(o). \tag{14}$$

For any sequence $o' = (o_0, \ldots, o_\ell)$, $|o'| \leq L + 1$, as the output of $\mathscr{A}(T, \mathcal{M}_s, \mathcal{M}_b)$. Then $o_{\ell-1}$ is the acceptance node in Decision Phase. We divide into three cases to prove the following equation

$$\mathbb{E}_{\mathcal{C}(v)} \left[\sum_{o' \subseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T]}{\mathcal{M}_b(o')} \right] = \sum_{o' \subseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T^{\text{sub}}]}{\mathcal{M}_b(o')}. \tag{15}$$

Case 1: the acceptance node $o_{\ell-1}$ is a preceding node of $\mathcal{C}(v)$. In this case, we obviously have

$$\Pr[\mathscr{A} = o' \mid T] = \Pr[\mathscr{A} = o' \mid T^{\text{sub}}]. \quad \forall o'$$

Then (15) always holds for this case.

Case 2: the acceptance node $o_{\ell-1} \in \mathcal{C}(v) \cup \{v\}$. Since Case 1 occurs with the same probability for both the trees T and T^{sub}, we only need to consider the conditional probability under the condition that Case 1 does not occur.

If $o_{\ell-1} \in \mathcal{C}(v)$, then the output sequence o' of $\mathscr{A}$ is the long chain in tree T (since the node v is the deepest parent node of T). Thus, $\ell = L$ and the length $|o'| = L + 1$. Note that $|o| = L + 1$, then for all sequences $o \supseteq o'$ (i.e., $o = o'$), the probability

$$\sum_{o' \subseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T]}{\mathcal{M}_b(o')} = p_v(o_{L-1}) \mathcal{M}_b(o_L \mid o_{[:L-1]}), \ (o_{L-1} \in \mathcal{C}(v)).$$

If $o_{\ell-1} = v$ (i.e., $o' = o_{[:L-1]}$ and $\ell = L - 1$), then the probability

$$\begin{aligned}
&\sum_{o' \subseteq o} \frac{\mathcal{M}_b(o) \Pr[\mathscr{A} = o' \mid T]}{\mathcal{M}_b(o')} \\
&= \Pr[\mathscr{A} = o' \mid T] \mathcal{M}_b(o_L \mid o') \\
&= p_v(o_{L-1}) \mathcal{M}_b(o_L \mid o_{[:L-1]}), \ (o_{L-1} \notin \mathcal{C}(v)).
\end{aligned}$$

To sum up, by the locally lossless of T, we know the Equation (15)

$$\begin{aligned}
&\mathbb{E}_{\mathcal{C}(v)}\left[\sum_{o'\subseteq o}\frac{\mathcal{M}_b(o)\Pr[\mathscr{A}=o'\mid T]}{\mathcal{M}_b(o')}\right]\\
&=\mathcal{M}_b(o_L\mid o_{[:L-1]})\cdot\mathbb{E}_{\mathcal{C}(v)}[p_v(o_{L-1})]\\
&\overset{(8)}{=}\mathcal{M}_b(o_L\mid o_{[:L-1]})\cdot\tilde{p}_v\mathcal{M}_b(o_{L-1}\mid v)\\
&=\mathcal{M}_b(o_L\mid o_{[:L-1]})\Pr[\mathscr{A}=(v,o_{L-1})\mid T^{\mathrm{sub}}]\\
&=\sum_{o'\subseteq o}\frac{\mathcal{M}_b(o)\Pr[\mathscr{A}=o'\mid T^{\mathrm{sub}}]}{\mathcal{M}_b(o')}
\end{aligned}$$

holds for this case.

Case 3: the acceptance node $o_{\ell-1}$ is an after-node of v. Since after rejection at node v, the remaining parts of T and T^{sub} are identical under the Decision Phase. Therefore, we only need to prove that the expectation of Case 3 occurring for T equals to the probability for T^{sub} rejecting v, i.e., to show that

$$\mathbb{E}_{\mathcal{C}(v)}\Pr\left(\mathscr{A}\text{ rejects }v\mid T\right)=\Pr\left(\mathscr{A}\text{ rejects }v\mid T^{\mathrm{sub}}\right).$$

Observe that in the Decision Phase of T^{sub}, when the algorithm reaches node v (which is a leaf in T^{sub}), it accepts v with probability $\tilde{p}_v$ and rejects it with probability

$$\Pr\left(\mathscr{A}\text{ rejects }v\mid T^{\mathrm{sub}}\right)=1-\tilde{p}_v.$$

In contrast, for the original tree T, when the algorithm traverses the local subtree $T_v=\{v\}\cup\mathcal{C}(v)$, the probability of rejecting all nodes in T_v is:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{C}(v)}\Pr\left(\mathscr{A}\text{ rejects }v\mid T\right)\\
&=\mathbb{E}_{\mathcal{C}(v)}\left[\prod_{u\in\mathcal{C}(v)}(1-\tilde{p}_u)\cdot(1-\tilde{p}_v^{\mathrm{res}})\right]\\
&=\mathbb{E}_{\mathcal{C}(v)}\left[\frac{\left(1-\sum_{u\in\mathcal{C}(v)}p_v(u)\right)\cdot p_v(\neg v)}{1-\sum_{u\in\mathcal{C}(v)}p_v(u)}\right]\\
&=\mathbb{E}_{\mathcal{C}(v)}\left[p_v(\neg v)\right].
\end{aligned}$$

To avoid sacrificing the generality of this proof by using specific expressions (e.g., (4)–(7)), we solely rely on the identity $p_v(\neg v)=1-\sum_{u\in\Sigma}p_v(u)$ here. Together with the locally lossless property, we have

$$\mathbb{E}_{\mathcal{C}(v)}\left[p_v(\neg v)\right]=1-\sum_{u\in\Sigma}\mathbb{E}_{\mathcal{C}(v)}\left[p_v(u)\right]=1-\tilde{p}_v\sum_{u\in\Sigma}\mathcal{M}_b(u\mid v)=1-\tilde{p}_v.$$

Therefore, by the case analysis above, Equation (15) holds. Combining this with the observation that $\mathbb{E}_T[\cdot]=\mathbb{E}_{T^{\mathrm{sub}}}\left[\mathbb{E}_{\mathcal{C}(v)}[\cdot]\right]$, we can establish:

$$\begin{aligned}
&\mathbb{E}_T\left[\sum_{o'\subseteq o}\frac{\mathcal{M}_b(o)\Pr[\mathscr{A}=o'\mid T]}{\mathcal{M}_b(o')}\right]\\
&=\mathbb{E}_{T^{\mathrm{sub}}}\mathbb{E}_{\mathcal{C}(v)}\left[\sum_{o'\subseteq o}\frac{\mathcal{M}_b(o)\Pr[\mathscr{A}=o'\mid T]}{\mathcal{M}_b(o')}\right]\\
&\overset{(15)}{=}\mathbb{E}_{T^{\mathrm{sub}}}\left[\sum_{o'\subseteq o}\frac{\mathcal{M}_b(o)\Pr[\mathscr{A}=o'\mid T^{\mathrm{sub}}]}{\mathcal{M}_b(o')}\right]\overset{(14)}{=}\mathcal{M}_b(o).
\end{aligned}$$

□

D.3 Conditional Optimality of UniVer

Proof of Theorem 5. Given any non-leaf node $v \in T$ with fixed $\tilde{p}_v$ and children $\mathcal{C}(v) = \{u_1, \dots, u_m\}$ generated by Greedy sampling (i.e., $H_v = \{u_1, \dots, u_{m-1}\}$ deterministic and $u_m \sim \mathcal{M}_s^{\neg}(\cdot \mid v)$), the conditional acceptance rate for $\mathcal{C}(v)$ is

$$\begin{aligned}&\alpha^*_{\mathrm{UniVer}}(\tilde{p}_v \mathcal{M}_b(\cdot \mid v), \mathcal{M}_s(\cdot \mid v)) \\ &= \mathbb{E}_{\mathcal{C}(v)}\left[\sum_{j=1}^{m} p_v(u_j) \,\middle|\, \tilde{p}_v\right] = \sum_{j=1}^{m} \mathbb{E}_{\mathcal{C}(v)}\left[p_v(u_j) \mid \tilde{p}_v\right].\end{aligned}$$

For the deterministic tokens $t \in H_v$, by local losslessness (Definition 1),

$$\mathbb{E}_{\mathcal{C}(v)}\left[p_v(t) \mid \tilde{p}_v\right] = \tilde{p}_v \mathcal{M}_b(t \mid v). \tag{16}$$

For the sampled node u_m, via Eq. (4),

$$\begin{aligned}\mathbb{E}_{\mathcal{C}(v)}\left[p_v(u_m) \mid \tilde{p}_v\right] &= \sum_{u_m \in \Sigma} \mathcal{M}_s^{\neg}(u_m \mid v) p_v(u_m) \\ &= \sum_{u \in \Sigma} \min\{\tilde{p}_v \mathcal{M}_b(u \mid v), \mathcal{M}_s^{\neg}(u \mid v)\}.\end{aligned}$$

Therefore, the conditional acceptance rate of UniVer equals Eq. (10).

To show optimality, note that under the constraint of local losslessness, the marginal $\mathbb{E}_{\mathcal{C}(v)}[p_v(t) \mid \tilde{p}_v] = \tilde{p}_v \mathcal{M}_b(t \mid v)$ is fixed for all $t \in \Sigma$. The only freedom is in allocating the mass for the sampled node u_m. To maximize $\mathbb{E}_{\mathcal{C}(v)}[p_v(u_m) \mid \tilde{p}_v]$, we solve the following scaled optimal transport problem conditioned on $\tilde{p}_v$:

$$\begin{aligned}&\max \sum_{u \in \Sigma} \mathcal{M}_s^{\neg}(u \mid v) p_v(u) \\ &\text{s.t. } \mathbb{E}_{\mathcal{C}(v)}[p_v(t) \mid \tilde{p}_v] = \tilde{p}_v \mathcal{M}_b(t \mid v), \quad \forall t \in \Sigma.\end{aligned}$$

Equivalently, letting $\gamma_v(u, t) := \mathcal{M}_s^{\neg}(u \mid v) p_v(t)$ for $u \in \Sigma, t \in \Sigma^+ := \Sigma \cup \{\neg v\}$, the Scaled-OT problem is

$$\begin{aligned}&\max \sum_{u \in \Sigma} \gamma_v(u, u) \\ &\text{s.t. } \sum_{u \in \Sigma} \gamma_v(u, \neg v) = 1 - \tilde{p}_v, \\ &\quad \sum_{u \in \Sigma} \gamma_v(u, t) = \tilde{p}_v \mathcal{M}_b(t \mid v), \quad \forall t \in \Sigma, \\ &\quad \sum_{t \in \Sigma^+} \gamma_v(u, t) = \mathcal{M}_s^{\neg}(u \mid v), \quad \forall u \in \Sigma, \\ &\quad \gamma_v(u, t) \geq 0, \quad \forall (u, t) \in \Sigma \times \Sigma^+.\end{aligned} \tag{17}$$

For the scaled optimal transport problem (17), the objective is to maximize the total mass on the diagonal $\sum_{u \in \Sigma} \gamma_v(u, u)$. By the marginal constraints, for each $u \in \Sigma$ we have:

$$\gamma_v(u, u) \leq \sum_{t \in \Sigma^+} \gamma_v(u, t) = \mathcal{M}_s^{\neg}(u|v), \tag{18}$$

and similarly for the column marginal:

$$\gamma_v(u, u) \leq \sum_{u' \in \Sigma} \gamma_v(u', u) = \tilde{p}_v \mathcal{M}_b(u|v). \tag{19}$$

Therefore,

$$\gamma_v(u, u) \leq \min\{\mathcal{M}_s^{\neg}(u \mid v), \tilde{p}_v \mathcal{M}_b(u \mid v)\}$$

for each u, yielding the upper bound:

$$\sum_{u \in \Sigma} \gamma_v(u, u) \leq \sum_{u \in \Sigma} \min\{\mathcal{M}_s^{\neg}(u \mid v), \tilde{p}_v \mathcal{M}_b(u \mid v)\}. \tag{20}$$

This bound is achieved by our UniVer, completing the proof of conditional optimality. □

D.4 Superiority of UniVer over Greedy Method

Throughout this subsection, $N_{\mathscr{A}}(T)$ denotes the number of accepted draft tokens produced by verification algorithm $\mathscr{A}$ on the draft tree T, excluding the root token and the final token sampled from the target model. All expectations are taken over the randomness in draft-tree generation.

We assume the standard subtree factorization induced by autoregressive drafting: conditioned on a node prefix v, the randomness used to generate the proper descendants of v is independent of the randomness outside the subtree T_v.

For clarity, we write $N_{\mathrm{Mod}}(T;\rho)$ for the acceptance length of modified UniVer when the root of T is assigned effective acceptance probability $\rho \in [0,1]$. Standard UniVer corresponds to $\rho = 1$.

Lemma 1 (Scaled Superiority)**.** *For any finite draft tree T with root r and any root effective acceptance probability $\rho \in [0,1]$, we have*

$$\mathbb{E}_T[N_{\mathrm{Mod}}(T;\rho)] \geq \rho\, \mathbb{E}_T[N_{\mathrm{Greedy}}(T)]\,. \tag{21}$$

Proof. We proceed by induction on the number of parent nodes n in T.

Base case ($n = 1$). The tree consists only of the root r and its children $\mathcal{C}(r)$. Therefore,

$$\begin{aligned}\mathbb{E}_T[N_{\mathrm{Mod}}(T;\rho)] &= \alpha^*_{\mathrm{UniVer}}\big(\rho\mathcal{M}_b(\cdot \mid r), \mathcal{M}_s(\cdot \mid r)\big)\\ &\geq \rho\, \alpha^*_{\mathrm{Greedy}}\big(\mathcal{M}_b(\cdot \mid r), \mathcal{M}_s(\cdot \mid r)\big)\\ &= \rho\, \mathbb{E}_T[N_{\mathrm{Greedy}}(T)]\,.\end{aligned}$$

The inequality follows directly from Eq. (2) and Eq. (10). In particular, for every $a, b \geq 0$ and $\rho \in [0,1]$,

$$\min\{\rho a, b\} \geq \rho \min\{a, b\}.$$

Inductive step. Assume that Lemma 1 holds for all draft trees with at most k parent nodes. Consider a tree T with $k+1$ parent nodes, where $k \geq 1$.

Let

$$\mathcal{C}(r) = \{v_1, \ldots, v_m\}$$

be the ordered children of the root, where v_m is the sampled child. Since T has at least two parent nodes, at least one child of r is a non-leaf node. Let v_j be the first non-leaf child in this order. Thus, $v_1, \ldots, v_{j-1}$ are leaves.

Define:

- T_{v_j} as the subtree rooted at v_j, including v_j and all its descendants;
- T^{new} as the tree obtained from T by removing all proper descendants of v_j, so that v_j becomes a leaf.

Both T_{v_j} and T^{new} contain at most k parent nodes.

Let

$$H_j := \prod_{\ell=1}^{j-1} (1 - \tilde{p}_{v_\ell})\,. \tag{22}$$

Since $v_1, \ldots, v_{j-1}$ are leaves, H_j is the conditional probability that the Decision Phase reaches the subtree rooted at v_j.

For any prefix token v and any $\eta \in [0,1]$, define

$$U(v,\eta) := \mathbb{E}_{T_v}[N_{\mathrm{Mod}}(T_v;\eta) \mid v]\,, \tag{23}$$

$$G(v) := \mathbb{E}_{T_v}[N_{\mathrm{Greedy}}(T_v) \mid v]\,. \tag{24}$$

Because the root token of a subtree is not counted in its acceptance length, replacing the leaf v_j in T^{new} by the full subtree T_{v_j} adds exactly the accepted descendants of v_j. Therefore,

$$\begin{aligned}\mathbb{E}_T[N_{\mathrm{Mod}}(T;\rho)] = \mathbb{E}_{T^{\mathrm{new}}}[N_{\mathrm{Mod}}(T^{\mathrm{new}};\rho)]\\ + \mathbb{E}_{T^{\mathrm{new}}}\big[H_j\, U\big(v_j, \tilde{p}_{v_j}\big)\big]\,.\end{aligned} \tag{25}$$

The value $\tilde{p}_{v_j}$ may depend on the root-level allocation randomness contained in T^{new}. However, after conditioning on T^{new}, it is fixed. Since the induction hypothesis holds for every root effective probability in $[0, 1]$, it can be applied pointwise to T_{v_j}:

$$U\big(v_j, \tilde{p}_{v_j}\big) \geq \tilde{p}_{v_j}\, G(v_j). \tag{26}$$

Applying the induction hypothesis also to T^{new} and substituting Eq. (26) into Eq. (25), we obtain

$$\begin{aligned}\mathbb{E}_T[N_{\text{Mod}}(T;\rho)] \geq \rho\, \mathbb{E}_{T^{\text{new}}}[N_{\text{Greedy}}(T^{\text{new}})] \\ + \mathbb{E}_{T^{\text{new}}}\left[H_j \tilde{p}_{v_j} G(v_j)\right].\end{aligned} \tag{27}$$

By construction of the conditional hazards in the Decision Phase,

$$p_r^{\text{Mod}}(v_j) = H_j \tilde{p}_{v_j}, \tag{28}$$

where $p_r^{\text{Mod}}(v_j)$ denotes the marginal probability mass allocated by modified UniVer to the root child v_j. Hence Eq. (27) becomes

$$\begin{aligned}\mathbb{E}_T[N_{\text{Mod}}(T;\rho)] \geq \rho\, \mathbb{E}_{T^{\text{new}}}[N_{\text{Greedy}}(T^{\text{new}})] \\ + \mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Mod}}(v_j) G(v_j)\right].\end{aligned} \tag{29}$$

The corresponding decomposition for Greedy is

$$\begin{aligned}\mathbb{E}_T[N_{\text{Greedy}}(T)] = \mathbb{E}_{T^{\text{new}}}[N_{\text{Greedy}}(T^{\text{new}})] \\ + \mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Greedy}}(v_j) G(v_j)\right],\end{aligned} \tag{30}$$

where $p_r^{\text{Greedy}}(v_j)$ is the marginal root-level acceptance probability of v_j under Greedy.

It remains to prove

$$\mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Mod}}(v_j) G(v_j)\right] \geq \rho\, \mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Greedy}}(v_j) G(v_j)\right]. \tag{31}$$

We distinguish two cases.

Case 1: v_j is a deterministic child. Suppose

$$v_j \in \{v_1, \ldots, v_{m-1}\}.$$

The token value of v_j is fixed. By the subtree-factorization assumption, conditioned on v_j, the randomness used to generate T_{v_j} is independent of the root-level allocation randomness. Consequently, $G(v_j)$ is constant with respect to the expectation over the root-level randomness in T^{new}.

By local losslessness of modified UniVer,

$$\mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Mod}}(v_j)\right] = \rho\, \mathcal{M}_b(v_j \mid r). \tag{32}$$

Similarly, the single-layer Greedy verifier is lossless and therefore

$$\mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Greedy}}(v_j)\right] = \mathcal{M}_b(v_j \mid r). \tag{33}$$

It follows that

$$\begin{aligned}\mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Mod}}(v_j) G(v_j)\right] &= G(v_j)\, \mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Mod}}(v_j)\right] \\ &= G(v_j) \cdot \rho\, \mathcal{M}_b(v_j \mid r) \\ &= \rho\, G(v_j)\, \mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Greedy}}(v_j)\right] \\ &= \rho\, \mathbb{E}_{T^{\text{new}}}\left[p_r^{\text{Greedy}}(v_j) G(v_j)\right].\end{aligned}$$

Case 2: $v_j = v_m$ is the sampled child. Let

$$q(u) := \mathcal{M}_s^{\top}(u \mid r).$$

For every u in the support of q, modified UniVer assigns

$$p_r^{\mathrm{Mod}}(u) = \min\left\{1, \frac{\rho\, \mathcal{M}_b(u \mid r)}{q(u)}\right\}, \tag{34}$$

whereas Greedy assigns

$$p_r^{\mathrm{Greedy}}(u) = \min\left\{1, \frac{\mathcal{M}_b(u \mid r)}{q(u)}\right\}. \tag{35}$$

Since $\rho \in [0, 1]$,

$$p_r^{\mathrm{Mod}}(u) \geq \rho\, p_r^{\mathrm{Greedy}}(u) \qquad \text{for every } u \in \mathrm{supp}(q). \tag{36}$$

Moreover, $G(u) \geq 0$. Therefore,

$$\begin{aligned}
\mathbb{E}_{T^{\mathrm{new}}}\left[p_r^{\mathrm{Mod}}(v_m) G(v_m)\right] &= \sum_{u \in \mathrm{supp}(q)} q(u)\, p_r^{\mathrm{Mod}}(u)\, G(u) \\
&\geq \rho \sum_{u \in \mathrm{supp}(q)} q(u)\, p_r^{\mathrm{Greedy}}(u)\, G(u) \\
&= \rho\, \mathbb{E}_{T^{\mathrm{new}}}\left[p_r^{\mathrm{Greedy}}(v_m) G(v_m)\right].
\end{aligned}$$

Thus Eq. (31) holds in both cases. Combining Eq. (29) with Eq. (30), we obtain

$$\begin{aligned}
\mathbb{E}_T[N_{\mathrm{Mod}}(T; \rho)] &\geq \rho\, \mathbb{E}_{T^{\mathrm{new}}}[N_{\mathrm{Greedy}}(T^{\mathrm{new}})] \\
&\quad + \rho\, \mathbb{E}_{T^{\mathrm{new}}}\left[p_r^{\mathrm{Greedy}}(v_j) G(v_j)\right] \\
&= \rho\, \mathbb{E}_T[N_{\mathrm{Greedy}}(T)].
\end{aligned}$$

This completes the induction. □

Proof of Theorem 6. Standard UniVer corresponds to root effective acceptance probability $\rho = 1$. Therefore, Lemma 1 gives

$$\mathbb{E}_T[N_{\mathrm{UniVer}}(T)] = \mathbb{E}_T[N_{\mathrm{Mod}}(T; 1)] \geq \mathbb{E}_T[N_{\mathrm{Greedy}}(T)].$$

Hence UniVer is never worse than Greedy in expected acceptance length. □

E Broader Impacts

This paper proposes UniVer, a novel verification mechanism which improves the acceptance length and throughput of speculative decoding. Our contribution mainly lies in the speculative decoding algorithm, which may boost the deployment of LLMs on resource-constrained devices such as computers, phones, and various edge devices. LLMs themselves may be applied to a wide range of scenarios and lead to various impacts, while this work does not directly contribute to them.

F Licenses of Artifacts

The artifacts used in this paper include the target and draft models, the EAGLE codebase, and the Spec-Bench benchmark. The Vicuna v1.3 target models are released under a non-commercial research license. The Llama 3.1 and Llama 2 target models are subject to their respective Community License Agreements. Qwen2-7B-Instruct is released under Apache-2.0.

The repositories for the EAGLE draft checkpoints used with Vicuna v1.3, Llama 3.1, and Qwen2-7B-Instruct list Apache-2.0 as their license. Use of each target model remains subject to its own license. For the Llama 2 experiments, we use Llama-68M as the draft model. Its model card lists Apache-2.0.

The EAGLE codebase and the Spec-Bench framework are released under Apache-2.0. The constituent datasets in Spec-Bench retain their original licenses.

G Use of AI Assistants

During the preparation of this work, the authors used AI assistants for portions of the code implementation and language polishing of the manuscript. All final content and technical contributions were reviewed and verified by the authors.

H Computing Resources

All experiments were conducted using NVIDIA RTX A6000 (48GB) GPUs. For models with 7B/8B and 13B parameters, each independent evaluation run requires approximately 30 to 60 minutes on a single GPU. For the larger 33B parameter models, an independent run takes under 2 hours utilizing two GPUs. We estimate the total computational budget for all experiments, including ablations and baselines, to be approximately 500 GPU hours.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We confirm that the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the theoretical guarantees and proofs in Section 4 and Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the implementation details and experimental setup in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code is proprietary currently.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the implementation details and experimental setup in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the error bars (mean±std) in the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the details in Section 5 and Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss societal impacts in Appendix E.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide information of existing assets in Appendix F.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We provide information of LLM use in Appendix G.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.