SHAPING WHAT TO VERIFY: REALIZABLE MULTI-PATH SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-path speculative decoding samples several draft continuations, but efficient lossless verification typically selects only one. This makes the selector crucial: it not only chooses a path, but also reshapes the proposal distribution seen by the verifier. We introduce *Sequential Martingale Selection* (SMS), a framework for optimizing this proposal while guaranteeing finite-sample realizability. SMS uses retained mass as the exact block-efficiency objective, represents selector-induced reweighting through locally conserved bounded densities, and realizes every proposal in the family with an explicit sequential selector. We further develop SMS-LP1, an online retained-flow optimizer that requires no additional target-model evaluation. Extensive experiments across LLM settings show that SMS-LP1 improves end-to-end decoding throughput by 20.8% over tokenwise verification and 14.7% over GBV on average. SMS-LP1 further achieves a 6.1% average gain on DFlash2, demonstrating substantial and consistent improvements on both standard model configurations and state-of-the-art drafting frameworks.

1 INTRODUCTION

Large language models (LLMs) generate autoregressively, making decoding inherently sequential and limiting parallelism (Radford et al., 2018; Brown et al., 2020; Chowdhery et al., 2023). Speculative decoding accelerates generation by letting a lightweight draft model propose future tokens and using the target model to verify them in parallel (Leviathan et al., 2023; Chen et al., 2023). Drafting determines the quality of candidate continuations, while verification determines how much useful output can be extracted from each expensive target-model invocation. This work focuses on the latter.

Verification methods have progressively increased how much draft computation can be converted into accepted output. Standard speculative decoding verifies a draft sequentially and stops retaining it after the first rejection, making all subsequent draft tokens unusable. Block Verification (BV) (Sun et al., 2025) and Hierarchical Speculative Decoding (HSD) (Zhou et al., 2026) jointly verify a complete draft path and can retain a longer prefix. Multi-path methods further sample K alternative draft paths, increasing the chance of finding a path compatible with the target for acceptance. However, verifying all K paths would also multiply expensive target-model work, potentially offsetting the benefit of additional candidates. GBV addresses this tension through a useful abstraction: a *selector* first chooses one of the K draft paths, and BV then verifies the proposal distribution induced by this selection rule (Thomas & Pal, 2026). The selector therefore does more than pick a promising candidate—it reshapes the proposal distribution actually seen by BV. A better selector can exploit multi-path diversity while applying Block Verification only to the selected path under the induced proposal. Yet practical selectors remain largely tied to fixed ranking rules, leaving this proposal-shaping space substantially underexplored.

Turning selector design from a fixed heuristic into systematic optimization raises three fundamental challenges. *First, what should a selector optimize?* A path likely under the target model may still be poor for block verification: an early mismatch makes all later tokens useless. So selector quality must reflect how probability mass survives across the prefix structure, rather than token-level similarity or a path-level score. *Second, which proposals are realizable?* As the selector must return one of only K sampled paths, it cannot arbitrarily reshape the draft distribution; prefix probabilities are coupled by global finite-sample constraints. Ignoring them may yield a desirable proposal that no valid selector can realize. *Third, how can such a proposal be optimized efficiently?* The global optimum depends

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

on target behavior at future prefixes, but obtaining it requires extra target-model evaluations that may erase the verification gains. So a practical selector must be *globally meaningful*, *exactly realizable*, and *locally computable* from information available during decoding. Together, these requirements turn multi-path verification from fixed ranking into principled proposal optimization.

Contributions. We introduce *Sequential Martingale Selection* (SMS), which addresses selector design in three steps: retained mass defines *what to optimize*, bounded martingales characterize *what can be realized*, and retained-flow optimization determines *how to optimize it efficiently online*.

First, to tell what to optimize, we introduce *retained mass*, which tracks the probability mass that both reaches and survives each prefix. After showing that BV and the theoretical formulation of HSD induce the same single-path verification law, we derive a simple supply–capacity recursion: surviving mass is jointly limited by the selected proposal and the continuation supported by the target. Crucially, total retained mass is exactly the block efficiency. This turns selector design from heuristic ranking into direct optimization of the quantity that verification ultimately retains.

Second, to ensure finite-K realizability, we represent selection as prefix-wise amplification of the draft distribution. These factors form a density martingale under a local conservation law. Bounding this martingale yields an explicit sequential gate that realizes every proposal in the family from the K sampled paths while preserving lossless BV. Despite this local construction, the family remains provably expressive: its optimum captures at least half of the best possible improvement over single-path BV for $K = 2$, approaching 63.2% as K grows.

Third, for higher online optimization efficiency, we develop *SMS-LP1* for efficient online optimization. At each visited prefix, it reallocates proposal flow toward continuations with unsaturated target-supported capacity while preserving finite-K realizability. SMS-LP1 optimizes only the next-layer retained mass, avoiding target evaluation at unseen descendants; when multiple local optima exist, a minimum-movement rule selects the one that least perturbs the original draft flow. The resulting method requires no additional target-model evaluation and remains exactly lossless.

Finally, SMS-LP1 yields substantial gains in both block efficiency and end-to-end decoding throughput across LLM and dLLM settings. For LLMs, across Qwen2.5-0.5B-Instruct paired with Qwen2.5-7B/14B/32B-Instruct targets and Llama-3.2-1B-Instruct paired with Llama-3.1-8B-Instruct, SMS-LP1 improves block efficiency and throughput by 32.5% and 20.8% over tokenwise verification on average, and by 10.9% and 14.7% over GBV. For dLLMs, using the state-of-the-art DFlash2 drafting framework, SMS-LP1 further improves block efficiency and throughput by 13.3% and 6.1% over its rejection-sampling verifier.

2 Related Work

Drafting Methods. A broad line of work improves speculative decoding by generating stronger or more efficient draft candidates, including multi-head prediction (Cai et al., 2024), auxiliary drafting (Li et al., 2024a;b; 2025), distillation (Zhou et al., 2024), recurrent and retrieval-based drafting (Cheng et al., 2024; He et al., 2024), and self-speculation (Elhoushi et al., 2024). Recent methods further align draft training with speculative verification, including GRIFFIN, GTO, and VSD (Hu et al., 2025; 2026; Zou et al., 2026), while SMART adapts speculative-tree expansion (Wang & Zhou, 2026). These methods improve candidate generation; our work instead focuses on how multiple sampled paths are selected and verified.

Single-Path Verification. Verification methods instead improve how draft candidates are accepted. Standard speculative decoding uses sequential token-wise acceptance rules (Leviathan et al., 2023; Chen et al., 2023), while Tree Monte Carlo (Hu & Huang, 2024), Block Verification (BV) (Sun et al., 2025), and Hierarchical Speculative Decoding (HSD) (Zhou et al., 2026) develop lossless sequence-level alternatives for the single-path setting. A complementary line of work relaxes exact target-distribution preservation for additional efficiency (Wang et al., 2026). In contrast, we focus on exact, lossless verification and revisit the relationship between BV and HSD in Sec. 4.1.

Multi-Path Verification. Multi-path verification exploits multiple candidate continuations through optimal-transport or tree-based formulations, including SpecTr (Sun et al., 2023), SpecInfer (Miao et al., 2024), Sequoia (Chen et al., 2024), and Traversal Verification (Weng et al., 2025). More recently, GBV (Thomas & Pal, 2026) selects one draft path through a global ranking before applying BV, while

SpecTr-GBV (Lin et al., 2026) combines SpecTr with Greedy-BV for multi-draft block verification. Our work builds on the selector–BV decomposition of GBV, but moves beyond a fixed ranking rule by optimizing a locally parameterized family of finite-K realizable proposal distributions.

3 PRELIMINARIES

Setup and notation. Consider one speculative-decoding iteration after a fixed context. Let p and q denote the target- and draft-model distributions, respectively. We use r for the *proposal distribution actually seen by the verifier*. Thus, $r = q$ in ordinary single-path decoding, while in the multi-path setting below $r = q^\Gamma$, the distribution induced by a path selector Γ. Throughout this work, all ratios are used only with positive denominators; otherwise, we use the equivalent probability-mass form.

For a block length L, let $X_{1:L} \sim r$ be the proposal presented to the verifier. If τ draft tokens are retained and one additional token is generated, the block efficiency is

$$\mathrm{BE}(r) := \mathbb{E}[\tau + 1]. \tag{1}$$

A larger BE means that each target-model invocation produces more output tokens on average.

Single-path block verification. BV (Sun et al., 2025) jointly verifies a complete proposal block. Its state $w_r(s)$ is the conditional probability that a proposed prefix s remains retainable. For a child sx,

$$w_r(sx) = \min\{1,\, w_r(s)p(x \mid s)/r(x \mid s)\}, \qquad w_r(\varnothing) = 1. \tag{2}$$

Intuitively, a child remains fully retainable when the target provides enough probability relative to the proposal; otherwise its retained probability is reduced accordingly. BV is lossless for any valid proposal r. HSD (Zhou et al., 2026) derives a related verifier from capped prefix likelihood ratios. Sec. 4.1 theoretically shows the same single-path verification law between BV and HSD.

Multi-path verification. It draws K independent paths $X_{1:L}^{(1)}, \ldots, X_{1:L}^{(K)} \sim q$ and uses a selector Γ to return one candidate. Because the selector favors some paths over others, the selected path follows a new distribution q^Γ, rather than the original draft distribution q. Following GBV (Thomas & Pal, 2026), the procedure can therefore be written as

$$K \text{ paths from } q \;\longrightarrow\; \Gamma \;\longrightarrow\; X^\star \sim q^\Gamma \;\longrightarrow\; \mathrm{BV}(p, q^\Gamma). \tag{3}$$

So once Γ is fixed, multi-path verification reduces to ordinary BV with a reshaped proposal distribution. The key question is therefore not only which sampled path to select, but which proposal q^Γ the selector should induce. The next section develops an exact objective for answering this question.

4 RETAINED MASS: WHAT SHOULD THE SELECTOR OPTIMIZE?

Sec. 3 shows that a multi-path selector acts by reshaping the proposal distribution seen by BV. So we first ask what makes one proposal better than another. Token-level agreement is insufficient because a later token is useful only if its preceding prefix also survives. We first unify existing single-path verifiers and then derive an exact objective for survival over the entire block.

4.1 A COMMON VIEW OF SINGLE-PATH VERIFICATION

Consider a realized proposal path $X_{1:L} \sim r$. At depth t, define

$$R_0 = 1, \qquad R_t := p(X_{1:t})/r(X_{1:t}), \qquad M_t := \max_{0 \le j \le t} R_j.$$

Here, R_t is the prefix likelihood ratio, measuring the target probability of the observed prefix relative to the proposal, while M_t records the largest such ratio encountered so far. As shown below, M_t captures the historical bottleneck that determines how much of the current prefix remains retainable.

Theorem 1 (BV–HSD Equivalence). *Define $w_t = w_r(X_{1:t})$. Along every realized proposal path,*

$$w_t = R_t/M_t, \qquad R_t^\star = R_t/M_{t-1} = w_{t-1} \cdot p(X_t \mid X_{<t})/r(X_t \mid X_{<t}), \tag{4}$$

where $R_t^\star$ is the capped prefix ratio used by HSD. So the theoretical formulation of HSD and BV induce the same prefix-acceptance probabilities, correction distributions, and accepted-prefix law.

See its proof in Appendix A.1. Theorem 1 indicates that BV and theoretical HSD are two parameterizations of the same single-path verification process, and thus are equivalent. So we can use the BV state $w_r(s)$ as a canonical description of whether a proposed prefix remains retainable. The distinct full-block rule used in the released HSD implementation is discussed separately in Appendix A.2.

4.2 Retained Mass as Exact Block Utility

The state $w_r(s)$ answers only a conditional question: *if the proposal reaches prefix s, how likely is it to survive?* Yet a highly survivable prefix contributes little if it is rarely proposed. Selector design must therefore also account for *how often s is reached*. We define the *retained mass* of prefix s as

$$\mu_r(s) := r(s)w_r(s), \qquad \mu_r(\varnothing) = 1. \tag{5}$$

Thus, $\mu_r(s)$ is the joint probability that the proposal reaches s and that s survives verification. Using the BV update, retained mass follows a simple local recursion:

$$\mu_r(sx) = \min\{r(sx),\, \mu_r(s)p(x \mid s)\}\,. \tag{6}$$

Intuitively, Eq. (6) says the proposal supplies at most $r(sx)$ mass to child sx, whereas the retained parent and target continuation support at most $\mu_r(s)p(x \mid s)$; only the smaller amount can survive.

Proposition 2 (Retained Mass Equals Block Efficiency)**.** *Define $J(r) := \sum_{s\in\mathcal{V}^{\le L}} \mu_r(s)$. Then*

$$J(r) = \sum\nolimits_{s\in\mathcal{V}^{\le L}} \mu_r(s) = 1 + \mathbb{E}[\tau] = \mathrm{BE}(r). \tag{7}$$

See its proof in Appendix A.3. Proposition 2 shows that at each depth, total retained mass is the probability that verification reaches that depth; summing these survival probabilities gives the expected retained length. So retained mass answers the first question in Sec. 1: a good selector places proposal mass on prefixes that can continue to survive, rather than merely on individually likely tokens or paths. The next question is which such proposals are realizable from only K candidates.

5 Sequential Martingale Selection: Which Proposals Are Realizable?

Retained mass in Sec. 4 identifies desirable proposals. Now we resolve which ones are realizable from only K sampled draft paths. Throughout this section we assume $K \ge 2$; $K = 1$ reduces to ordinary BV. As a selector must choose among these candidates, prefix probabilities are globally constrained by finite-sample availability. We define the complete finite-K selector space:

$$\mathcal{R}_K(q) := \left\{q^\Gamma : \Gamma \text{ is a randomized selector over } K \text{ i.i.d. paths from the draft distribution } q\right\}. \tag{8}$$

A selector cannot reshape q arbitrarily, as it must return an available candidate. GBV characterizes $\mathcal{R}_K(q)$ exactly via global antichain constraints (Thomas & Pal, 2026), recalled in Appendix B.1. Intuitively, probability can be assigned to a set of non-overlapping prefixes only when at least one sampled path enters that set. The resulting constraints couple all branches and all antichains, so a locally desirable change can violate feasibility elsewhere. SMS instead enforces finite-K realizability through locally propagated states with simple conservation and bounds.

5.1 From Global Feasibility to Local Density Conservation

The antichain characterization describes exactly which proposals are finite-K realizable, but it is global: feasibility of one prefix depends on how probability mass is distributed across many other branches. For online selector design, we instead need a state that can be propagated and optimized locally while remaining consistent with a valid selected-path distribution.

For a realizable proposal $r = q^\Gamma$ and a prefix with $q(s) > 0$, define its *prefix density*

$$A_s := r(s)/q(s), \qquad A_\varnothing = 1. \tag{9}$$

The value $A_s > 1$ means that selection amplifies prefix s relative to ordinary drafting, whereas $A_s < 1$ means that it suppresses it. Prefix consistency imposes the following local conservation law.

Lemma 3 (Prefix-Density Conservation)**.** *Every realizable proposal $r = q^\Gamma$ satisfies*

$$A_s = \sum\nolimits_{x\in\mathcal{V}} q(x \mid s)A_{sx}. \tag{10}$$

See its proof in Appendix B.2. Lemma 3 shows that parent amplification is the draft-weighted average of its children: increasing probability on some branches must be compensated elsewhere. So along $X_{1:L} \sim q$, the process $\{A_{X_{1:t}}\}_{t=0}^{L}$ is a nonnegative martingale, replacing a global distribution with a locally propagated state. Conservation alone, however, does not ensure finite-K realizability; some locally consistent densities may still require more prefix mass than K sampled paths can supply. We next impose simple bounds that make every such local density explicitly realizable.

5.2 Bounded Gates Give an Exact Finite-K Selector

For $m \in (0, 1)$, SMS introduces a bounded gate process satisfying the same local conservation rule

$$0 \le \alpha_s \le 1, \qquad \alpha_\varnothing = m, \qquad \alpha_s = \sum\nolimits_x q(x \mid s)\alpha_{sx}, \tag{11}$$

and maps it to the prefix density through

$$A_s = d_m + C_m\alpha_s, \qquad d_m := (1-m)^{K-1}, \qquad C_m := (1-d_m)/m. \tag{12}$$

Here, α_s measures how strongly continuations through prefix s are favored by the selector; under q, it is also the expected terminal gate of continuations below s. The affine mapping to A_s preserves this local conservation, while the bounds $0 \le \alpha_s \le 1$ keep the induced density finite and, with the chosen coefficients d_m and C_m, exactly realizable from K sampled paths. Operationally, randomly order the K candidates, sequentially select each of the first $K-1$ paths X with probability α_X if no earlier gate accepts, and use the final candidate as a fallback. The bounded gates define the proposal family

$$\mathcal{S}_{K,m}(q) := \{r_\alpha : r_\alpha(X) = q(X)(d_m + C_m\alpha_X),\ \{\alpha_s\} \text{ satisfies Eq. (11)}\}\,. \tag{13}$$

Thus, each member of $\mathcal{S}_{K,m}(q)$ reweights the original draft distribution through a bounded and locally conserved gate process, where $A_X = d_m + C_m\alpha_X$ is the amplification applied to path X. The key question is whether every such locally specified proposal can actually be realized by selecting one of only K sampled paths. The results below give a positive answer with its proof in Appendix B.3.

Theorem 4 (Exact Finite-K Realization)**.** *The sequential gate above induces*

$$q^\Gamma(X) = q(X)(d_m + C_m\alpha_X) = q(X)A_X. \tag{14}$$

Consequently, for every prefix s with $q(s) > 0$,

$$q^\Gamma(s) = q(s)A_s, \qquad q^\Gamma(x \mid s) = q(x \mid s)A_{sx}/A_s. \tag{15}$$

Eq. (14) in Theorem 4 shows that the local density state A_X is exactly the factor by which selection changes the probability of path X. Eq. (15) shows that the induced proposal conditional can be computed locally: selection rescales $q(x \mid s)$ by the relative amplification A_{sx}/A_s. This quantity is precisely what BV needs for lossless verification. Hence, we have

$$\mathcal{S}_{K,m}(q) \subseteq \mathcal{R}_K(q).$$

Rather than optimizing globally coupled antichain constraints directly, SMS optimizes locally conserved and bounded densities; every feasible SMS construction is realized by a valid finite-K selector.

5.3 Coverage of the Full Selector Space

The local construction above guarantees realizability, but it could in principle be overly restrictive. We therefore ask how much expressive power is lost relative to the full finite-K selector space $\mathcal{R}_K(q)$. We set $m = 1/K$, which maximizes the universal coverage within this construction, and define

$$d_K := (1 - 1/K)^{K-1}, \qquad C_K := K(1 - d_K), \qquad \mathcal{S}_K(q) := \mathcal{S}_{K,1/K}(q). \tag{16}$$

Theorem 5 (Radial and Objective Coverage)**.** *The SMS family $\mathcal{S}_K(q)$ satisfies*

$$d_K q + (1 - d_K)\mathcal{R}_K(q) \subseteq \mathcal{S}_K(q) \subseteq \mathcal{R}_K(q). \tag{17}$$

Let $J^\star_{\mathcal{R}}$ and $J^\star_{\mathcal{S}}$ be the largest retained-mass objectives over $\mathcal{R}_K(q)$ and $\mathcal{S}_K(q)$, respectively. Then

$$J^\star_{\mathcal{S}} - J(q) \ge (1 - d_K)\big(J^\star_{\mathcal{R}} - J(q)\big). \tag{18}$$

See Appendix B.4 for the proof. Eq. (17) gives the geometric guarantee: from the base proposal q, SMS preserves every direction available to the full selector space, up to the common factor $1 - d_K$. Eq. (18) converts this coverage into performance: by concavity of retained mass, the best SMS proposal captures at least the same fraction of the maximum improvement over single-path BV. This fraction is $1/2$ for $K = 2$, $5/9$ for $K = 3$, and approaches $1 - e^{-1} \approx 63.2\%$. Thus, SMS answers the second question in Sec. 1: it replaces global finite-K constraints with local states that are easy to propagate, exactly realizable, and provably expressive. We next address the final question: how to choose these states online without additional target-model evaluation.

6 RETAINED-FLOW OPTIMIZATION: HOW TO OPTIMIZE ONLINE?

Having established a locally parameterized and finite-K realizable proposal family in Sec. 5, we now optimize its density states for retained mass. The full-horizon optimum, however, depends on target probabilities at unvisited prefixes, whose evaluation would undermine the intended speedup. We therefore cast proposal shaping as a local flow problem and optimize the retained-mass contribution observable at each visited prefix using only target information already available during decoding.

6.1 LOCAL RETAINED-FLOW OPTIMIZATION

Recall $q^\Gamma(s) = q(s)A_s$ in Eq. (15). To track how much of this mass actually survives BV, define

$$B_s := \mu(s)/q(s) \quad \rightarrow \quad \mu(s) = q(s)B_s. \tag{19}$$

Intuitively, A_s describes the proposal flow arriving at s, relative to draft probability $q(s)$, whereas B_s describes the portion that remains retained after verification. At prefix s, define

$$q_x := q(x \mid s), \qquad p_x := p(x \mid s), \qquad z_s(x) := q_x A_{sx}. \tag{20}$$

The quantity $z_s(x)$ is the normalized proposal flow sent from s to child sx: indeed, $q^\Gamma(sx) = q(s)z_s(x)$. The martingale conservation law requires the child flows to sum to the parent flow, while the SMS density bounds limit how much any child can be amplified. Hence,

$$\mathcal{Z}_s(A_s) := \{z \geq 0 : \sum\nolimits_x z_x = A_s,\ d_K q_x \leq z_x \leq (d_K + C_K)q_x\}. \tag{21}$$

Combining this flow with retained-mass recursion yields

$$\mu(sx) = q(s)\min\{z_s(x), B_s p_x\}. \tag{22}$$

Eq. (22) is the key local optimization view: the selector supplies $z_s(x)$, while the retained parent and target continuation support at most $B_s p_x$; only their overlap survives. The resulting child states are

$$A_{sx} = z_s(x)/q_x, \qquad B_{sx} = \min\{A_{sx}, B_s p_x/q_x\}. \tag{23}$$

Thus, a local allocation z_s determines both the proposal reaching each child and the retained mass available for subsequent layers. Optimizing all such allocations jointly would give the best proposal in $\mathcal{S}_K(q)$ defined in Eq. (16); its full-horizon Bellman formulation is given in Appendix C.1. However, its continuation values require target probabilities at unvisited descendants. Our practical method SMS-LP1 instead optimizes the exact retained mass contributed by the next layer.

$$\max\nolimits_{z \in \mathcal{Z}_s(A_s)} \sum\nolimits_{x \in \mathcal{V}} \min\{z_x, B_s p_x\}. \tag{24}$$

We call Eq. (24) the *one-layer retained-mass problem* (LP1) and denote its solution set by $\mathcal{Z}_s^\star$. LP1 reallocates proposal flow toward children with unsaturated target-supported capacity while preserving finite-K realizability. By Eq. (22), it exactly maximizes the next-layer contribution to J, without additional target evaluation. As a horizon-one objective, it does not optimize the unknown future subtree. Given a solution z_s, the induced proposal conditional needed by BV is obtained directly as

$$q^\Gamma(x \mid s) = z_s(x)/A_s, \qquad \alpha_{sx} = (A_{sx} - d_K)/C_K. \tag{25}$$

The resulting states $(A_{sx}, B_{sx}, \alpha_{sx})$ are then propagated along the sampled paths. LP1 can, however, have several allocations with identical next-layer utility but different downstream states.

6.2 MINIMUM-MOVEMENT COMPLETION

We resolve this ambiguity conservatively. Define

$$c_x := B_s p_x, \qquad \ell_x := d_K q_x, \qquad u_x := (d_K + C_K)q_x, \qquad z_x^0 := A_s q_x. \tag{26}$$

Here, c_x is the target-supported retained capacity, $[\ell_x, u_x]$ is the SMS realizability range, and $z^0 = (z_x^0)_{x \in \mathcal{V}}$ is the original draft-proportional flow. Because $A_s \in [d_K, d_K + C_K]$, $z^0 \in \mathcal{Z}_s(A_s)$. As shown in Appendix C.2, LP1 is equivalently the L_1 projection of c onto the feasible flow set.

$$\mathcal{Z}_s^\star = \arg\min\nolimits_{z \in \mathcal{Z}_s(A_s)} \|z - c\|_1.$$

Algorithm 1 SMS-LP1

Require: K draft paths $\{X^{(i)}_{1:L}\}_{i=1}^{K} \overset{\text{i.i.d.}}{\sim} q$, and their corresponding cached target distributions $p_s = p(\cdot \mid s)$ during verification and draft distributions $q_s = q(\cdot \mid s)$ during drafting

1: **if** $K = 1$, **then return** $\mathrm{BV}(X^{(1)}_{1:L}; p, q)$
2: $m \leftarrow 1/K$, $d_K \leftarrow (1-m)^{K-1}$, $C_K \leftarrow (1-d_K)/m$, and initialize $(A_\varnothing, B_\varnothing, \alpha_\varnothing) \leftarrow (1, 1, m)$
3: **for** $t = 0, \ldots, L-1$ **do**
4: **for** each distinct visited prefix $s = X^{(k)}_{1:t}$ **do**
5: Compute z_s from Eqs. (26)–(28)
6: Store $q^\Gamma(\cdot \mid s) \leftarrow z_s / A_s$, and update child states via Eqs. (23) and (25)
7: Randomly order the K complete candidates
8: Gate the first $K-1$ using terminal α_X; use the last as fallback
9: Let $X^\star$ be the selected path
10: **return** $\mathrm{BV}(X^\star; p, q^\Gamma)$

Among its possibly many solutions, we choose one that moves the least mass from z^0. Intuitively, if z^0_x already exceeds the useful capacity c_x, part of its flow can be removed without reducing retained mass. Conversely, if $z^0_x < c_x$, additional flow can improve retained mass until either the target capacity or the SMS upper bound is reached. Define these removable and useful amounts as

$$a^-_x := \left[z^0_x - \max\{\ell_x, c_x\}\right]_+, \qquad a^+_x := \left[\min\{u_x, c_x\} - z^0_x\right]_+. \tag{27}$$

Let $D^- := \sum_x a^-_x$, $D^+ := \sum_x a^+_x$, $\Delta := \min\{D^-, D^+\}$. The maximum useful transfer is therefore Δ. SMS-LP1 moves exactly this amount, proportionally across the available donors and receivers:

$$z^{\mathrm{MM}}_x = z^0_x - \Delta/D^- \cdot a^-_x + \Delta/D^+ \cdot a^+_x, \tag{28}$$

where a fraction is defined as zero when its denominator is zero.

Theorem 6 (Exact Minimum-Movement LP1)**.** *The allocation z^{MM} is feasible and exactly solves LP1. Moreover,*

$$z^{\mathrm{MM}} \in \arg\min_{z \in \mathcal{Z}^\star_s} \|z - z^0\|_1. \tag{29}$$

See its proof in Appendix C.3. Theorem 6 shows that (i) z^{MM} maximizes next-layer retained mass; and (ii) among all such optima, it minimally perturbs the original draft flow. Thus, minimum movement is a conservative tie-break rather than a lookahead objective: it does not predict unknown future target behavior, but avoids unnecessary redistribution when that information is unavailable.

Solver and complexity. The solution z^{MM} is obtained directly from Eqs. (26)–(28), without a generic LP solver. Given p_s, q_s, A_s, B_s, we compute the donor and receiver capacities a^- and a^+, their totals D^- and D^+, and then apply the closed-form update in Eq. (28). This requires $O(|\mathcal{V}|)$ operations per visited prefix, where $|\mathcal{V}|$ is the vocabulary size, with no sorting, iterative optimization, or additional model forward. The computation is fully vectorizable. With K paths of length L, the implementation performs $O(KL|\mathcal{V}|)$ vectorizable operations in the worst case.

6.3 Complete Algorithm and Losslessness

Algorithm 1 applies the local rule along the prefixes visited by the K draft paths. At each visited prefix, it solves the minimum-movement LP1 problem, updates the local states, and stores the induced conditional $q^\Gamma(\cdot \mid s)$. Since the update is prefix-local and deterministic, identical shared prefixes receive identical states. The terminal gates then invoke the sequential selector of Theorem 4 to choose one candidate, which is verified once by ordinary BV under the induced proposal q^Γ. This yields the exactness guarantee below.

Theorem 7 (Losslessness of SMS-LP1)**.** *Algorithm 1 induces a proposal $q^\Gamma \in \mathcal{S}_K(q) \subseteq \mathcal{R}_K(q)$. Applying BV to the selected path with this induced proposal preserves exactly target distribution p.*

See its proof in Appendix C.4. Theorem 7 proves the losslessness of SMS-LP1: the output can preserve the target distribution p. The reason is direct: LP1 changes only the local density states while remaining inside the SMS family; so the sequential gate realizes exactly the corresponding q^Γ, and BV is lossless for that proposal. SMS-LP1 needs only one ordinary BV call after selection, no additional target-model evaluation, and linear-time vocabulary operations at each visited prefix.

7 EXPERIMENTS

7.1 EXPERIMENTAL SETUP

Models and Benchmarks. We evaluate SMS-LP1 across different model families and target scales, including Qwen2.5-0.5B-Instruct with Qwen2.5-7B/14B/32B-Instruct targets (Yang et al., 2024) and Llama-3.2-1B-Instruct with Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We further evaluate Muse-Glimmer-30B-DFlash2 with Muse-Glimmer-30B to demonstrate compatibility with the state-of-the-art DFlash2 drafting framework (Chen et al., 2026; Inco AI, 2026). We evaluate mathematical reasoning, code generation, and conversation using GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), HumanEval (Chen et al., 2021), MBPP-sanitized (Austin et al., 2021), and MT-Bench (Zheng et al., 2023).

Baselines and Experimental Settings. We compare SMS-LP1 with three lossless baselines: *Token-wise* refers to standard speculative decoding, *BV* is single-path block verification and *GBV* denotes greedy multi-path verification. All methods are evaluated on the same hardware (NVIDIA RTX PRO 6000) and the same implementation framework for fair comparison. Unless otherwise specified, we use $K = 6$, $L = 8$, $T = 1$, and batch size one. We report block efficiency (BE) and end-to-end throughput (E2E, generated tokens/s) as evaluation metrics. Further details are given in Appendix D.

7.2 MAIN RESULTS

Results with Autoregressive Drafting. Table 1 shows that SMS-LP1 achieves the highest block efficiency and end-to-end throughput across all settings. On average, SMS-LP1 improves block efficiency and throughput by 32.5% and 20.8% over tokenwise verification, by 27.1% and 15.7% over BV, and by 10.9% and 14.7% over GBV.

The throughput gains remain substantial across task categories, with stronger improvements on more challenging benchmarks. On mathematical reasoning, SMS-LP1 improves E2E throughput by 17.3% on GSM8K and 21.0% on MATH-500. On code generation, the corresponding gains are 16.7% on HumanEval and 23.5% on MBPP. The strongest gain occurs on MT-Bench, reaching 28.9%. This pattern is consistent with higher generation uncertainty producing more diverse candidate paths and greater room for proposal shaping and retained-mass reallocation.

Table 1: Comparison of block efficiency (BE) and end-to-end throughput (E2E, tokens/s) on five benchmarks at temperature $T = 1$. Please refer to E.1 for the stability results

Model pair	Method	GSM8K		HumanEval		MBPP		MATH-500		MT-Bench		Average	
		BE	E2E	BE	E2E	BE	E2E	BE	E2E	BE	E2E	BE	E2E
Qwen2.5 0.5B → 7B	Tokenwise	5.47	49.83	5.28	49.09	4.81	44.70	5.67	52.60	3.37	31.23	4.92	45.49
	BV	5.59	50.63	5.48	51.12	4.85	45.03	5.78	53.98	3.55	33.04	$5.05_{+2.7\%}$	$46.76_{+2.8\%}$
	GBV	6.44	52.27	6.17	51.89	5.83	48.91	6.58	55.45	4.33	36.25	$5.87_{+19.3\%}$	$48.95_{+7.6\%}$
	SMS-LP1	**7.03**	**59.08**	**6.83**	**59.56**	**6.49**	**56.50**	**7.22**	**63.20**	**4.73**	**41.01**	$\mathbf{6.46}_{+31.3\%}$	$\mathbf{55.87}_{+22.8\%}$
Qwen2.5 0.5B → 14B	Tokenwise	5.11	40.07	4.95	42.20	4.33	35.71	5.29	44.39	3.12	27.44	4.56	37.96
	BV	5.31	41.84	5.09	41.20	4.59	37.83	5.54	46.08	3.27	26.77	$4.76_{+4.4\%}$	$38.74_{+2.1\%}$
	GBV	6.12	41.47	6.05	43.54	5.37	39.58	6.30	46.74	4.07	29.57	$5.58_{+22.4\%}$	$40.18_{+5.8\%}$
	SMS-LP1	**6.70**	**46.62**	**6.40**	**47.54**	**6.13**	**46.64**	**6.91**	**52.97**	**4.47**	**33.65**	$\mathbf{6.12}_{+34.2\%}$	$\mathbf{45.48}_{+19.8\%}$
Qwen2.5 0.5B → 32B	Tokenwise	5.06	30.56	4.84	31.30	4.62	30.24	5.31	34.18	3.13	20.30	4.59	29.32
	BV	5.26	33.31	5.10	34.14	4.78	31.16	5.53	37.02	3.27	21.98	$4.79_{+4.2\%}$	$31.52_{+7.5\%}$
	GBV	6.14	32.31	5.96	34.19	5.73	32.27	6.30	36.27	4.07	23.34	$5.64_{+22.8\%}$	$31.68_{+8.0\%}$
	SMS-LP1	**6.59**	**34.83**	**6.43**	**37.78**	**6.35**	**36.19**	**6.97**	**41.11**	**4.49**	**26.43**	$\mathbf{6.17}_{+34.2\%}$	$\mathbf{35.27}_{+20.3\%}$
Llama 3.2 1B → 3.1 8B	Tokenwise	5.18	58.20	6.53	76.36	5.86	68.86	5.44	63.84	4.26	48.29	5.45	63.11
	BV	5.50	61.45	6.82	79.50	6.07	71.01	5.79	67.69	4.70	53.15	$5.78_{+5.9\%}$	$66.56_{+5.5\%}$
	GBV	6.02	57.95	7.14	74.00	6.45	67.28	6.04	62.77	5.32	53.58	$6.19_{+13.6\%}$	$63.11_{+0.0\%}$
	SMS-LP1	**6.98**	**69.95**	**7.87**	**85.52**	**7.38**	**80.75**	**7.27**	**79.25**	**6.00**	**63.42**	$\mathbf{7.10}_{+30.2\%}$	$\mathbf{75.78}_{+20.1\%}$
DFlash2 → Glimmer-30B	Tokenwise	6.75	107.45	6.99	107.15	4.76	77.52	6.00	96.16	4.40	70.87	5.78	91.83
	BV	6.81	108.24	7.03	107.76	4.86	78.92	6.06	97.03	4.64	74.46	$5.88_{+1.8\%}$	$93.28_{+1.6\%}$
	GBV	7.16	108.37	7.20	105.83	4.98	75.08	6.47	97.59	4.89	72.38	$6.14_{+6.3\%}$	$91.85_{+0.0\%}$
	SMS-LP1	**7.38**	**111.30**	**7.63**	**112.03**	**5.74**	**85.98**	**6.75**	**101.14**	**5.23**	**76.93**	$\mathbf{6.55}_{+13.3\%}$	$\mathbf{97.48}_{+6.1\%}$

Table 2: Ablation studies on GSM8K with Qwen2.5-0.5B-Instruct as the draft model and Qwen2.5-32B-Instruct as the target model.

Setting		Tokenwise		BV		GBV		SMS-LP1	
Ablation	Value	BE	E2E	BE	E2E	BE	E2E	BE	E2E
K	1	5.06	30.56	5.26	33.31	5.26	33.31	5.26	33.31
	2	*N/A for single-path baselines*				5.60	30.71	5.83	32.08
	4					5.90	31.11	6.30	34.75
	6					6.14	32.31	6.59	**34.83**
	8					6.18	29.56	**6.77**	34.13
L	8	5.06	30.56	5.26	33.31	6.14	32.31	**6.59**	**34.83**
	12	5.77	27.65	6.21	29.66	7.36	29.29	**8.21**	**33.36**
	16	6.26	24.35	6.78	26.24	8.12	27.58	**9.26**	**32.17**
T	0.6	5.25	31.87	5.37	32.55	6.13	30.94	**6.34**	**32.67**
	0.8	5.17	30.57	5.36	31.60	6.18	30.42	**6.56**	**32.94**
	1.0	5.06	30.56	5.26	33.31	6.14	32.31	**6.59**	**34.83**

Results with Diffusion Drafting. SMS-LP1 remains effective with the state-of-the-art DFlash2 drafting framework. Compared with the tokenwise rejection-sampling baseline, SMS-LP1 improves block efficiency and end-to-end throughput by 13.3% and 6.1% on average, respectively, while further improving over GBV by 6.6% and 6.1%. The throughput gains are particularly pronounced on MBPP and MT-Bench, reaching 10.9% and 8.6%. These results show that SMS-LP1 provides additional verification gains on top of an advanced drafting framework without modifying the drafter.

7.3 Ablation Study

Effect of the Path Budget. We first vary the number of candidate paths K, where $K = 1$ reduces SMS-LP1 to the single-path BV baseline. As K increases, block efficiency improves steadily from 5.26 at $K = 1$ to 6.77 at $K = 8$, yielding a 28.6% gain and showing that SMS-LP1 can effectively exploit the additional selector freedom provided by larger path budgets. End-to-end throughput peaks at 34.83 tokens/s with $K = 6$, 4.6% above GBV's best observed 33.31 tokens/s at $K = 1$, before slightly decreasing at $K = 8$, suggesting that the overhead of larger path budgets eventually offsets the verification gains.

Effect of the Block Length. Across all tested block lengths, SMS-LP1 achieves the highest block efficiency and end-to-end throughput among the compared methods. As L increases from 8 to 16, its block efficiency rises from 6.59 to 9.26, corresponding to a 40.5% increase, while end-to-end throughput decreases from 34.83 to 32.17 tokens/s. Longer blocks provide more downstream positions over which retained-mass reallocation can improve verification efficiency, while the additional drafting and verification cost limits the corresponding throughput gain.

Effect of the Sampling Temperature. Across $T \in \{0.6, 0.8, 1.0\}$, SMS-LP1 consistently achieves the highest block efficiency and end-to-end throughput among the compared methods. As the temperature increases from 0.6 to 1.0, block efficiency rises from 6.34 to 6.59, while end-to-end throughput increases from 32.67 to 34.83 tokens/s. Higher temperatures increase diversity among sampled paths, providing more useful alternatives for retained-mass reallocation and allowing the selector to better exploit the multi-path budget.

8 Conclusion

We introduced SMS, a realizable proposal-shaping framework for multi-path speculative decoding. By combining retained-mass optimization, bounded martingale selection, and efficient local flow updates, SMS improves verification efficiency losslessly without additional target-model evaluation. Experiments show consistent gains over existing multi-path block-verification baselines.

Limitations and Scope. Our experiments focus on moderate path budgets, as larger candidate sets introduce increasing candidate-generation and selection overhead and are more costly to evaluate systematically. Our compatibility study currently focuses on DFlash2, extending SMS-LP1 to a broader range of advanced drafting methods is left for future work.

AI USE STATEMENT

Generative AI tools were used solely for grammar refinement and language polishing. All scientific ideas, theoretical developments, methodology, experiments, analysis, and conclusions are the authors' own contributions. AI-assisted edits were limited to improving readability, including grammar, spelling, sentence structure, and word choice, and were not used to modify the technical content or reported experimental results. All AI-assisted edits were manually reviewed by the authors to ensure that they preserved the intended meaning and technical accuracy of the manuscript.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732. 8

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf. 1

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5209–5235. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/cai24b.html. 2

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318. 1, 2

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block Diffusion for Flash Speculative Decoding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. 8

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374. 8

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. Sequoia: Scalable and robust speculative decoding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 129531–129563. Curran Associates, Inc., 2024. doi: 10.52202/079017-4116. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ea1f5f0878d43ff4fb8bf64ef4a2326c-Paper-Conference.pdf. 2

Yunfei Cheng, Aonan Zhang, Xuanyu Zhang, Chong Wang, and Yi Wang. Recurrent drafter for fast speculative decoding in large language models, 2024. URL https://arxiv.org/abs/2403.09919. 2

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. URL http://jmlr.org/papers/v24/22-1144.html. 1

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168. 8

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling early exit inference and self-speculative decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.681. URL https://aclanthology.org/2024.acl-long.681/. 2

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783. 8

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason Lee, and Di He. REST: Retrieval-based speculative decoding. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1582–1595, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.88. URL https://aclanthology.org/2024.naacl-long.88/. 2

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf. 8

Shijing Hu, Jingyang Li, Xingyu Xie, Zhihui Lu, Kim-chuan Toh, and Pan Zhou. GRIFFIN: Effective token alignment for faster speculative decoding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 125648–125673. Curran Associates, Inc., 2025. doi: 10.52202/085713-4188. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b6e67ae290635d0874c4cb43ba2a2cfb-Paper-Conference.pdf. 2

Shijing Hu, Jingyang Li, Zhihui Lu, and Pan Zhou. Bridging draft policy misalignment: Group tree optimization for speculative decoding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and

A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 112507–112524, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b6e67ae290635d0874c4cb43ba2a2cfb-Paper-Conference.pdf. 2

Zhengmian Hu and Heng Huang. Accelerated speculative sampling based on tree Monte Carlo. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19216–19251. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hu24f.html. 2

Inco AI. DFlash 2: Keep Drafting Parallel, August 2026. URL https://inco.ai/blog/dflash2/. 8

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html. 1, 2

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/li24bt.html. 2

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7421–7432, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.422. URL https://aclanthology.org/2024.emnlp-main.422/. 2

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4562. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c7b5a35ea98b62512a869c19ea7b03cb-Paper-Conference.pdf. 2

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf. 8

Yijun Lin, Jinhao Sheng, Qingyue Cai, and Feng Zhou. Spectr-gbv: Multi-draft block verification accelerating speculative decoding, 2026. URL https://arxiv.org/abs/2604.25925. 3, 23

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. SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification. In Rajiv Gupta, Nael B. Abu-Ghazaleh, Madan Musuvathi, and Dan Tsafrir (eds.), *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949. ACM, 2024. doi: 10.1145/3620666.3651335. URL https://doi.org/10.1145/3620666.3651335. 2

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. OpenAI Technical Report, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. 1

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. Spectr: Fast speculative decoding via optimal transport. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 30222–30242. Curran Associates, Inc., 2023. doi: 10.52202/075280-1314. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6034a661584af6c28fd97a6f23e56c0a-Paper-Conference.pdf. 2

Ziteng Sun, Uri Mendlovic, Yaniv Leviathan, Asaf Aharoni, Jae H Ro, Ahmad Beirami, and Ananda Theertha Suresh. Block verification accelerates speculative decoding. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 24965–24994, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3e710b42b1a9ed898f607ec0f4fcc971-Paper-Conference.pdf. 1, 2, 3

Rahul Thomas and Arka Pal. Greedy multi-path block verification for faster decoding in speculative sampling, 2026. URL https://arxiv.org/abs/2602.16961. 1, 2, 3, 4, 17

Lifu Wang and Pan Zhou. SMART: When is it actually worth expanding a speculative tree? In *Computer Vision – ECCV 2026*, pp. 228–244, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37627-5. doi: 10.1007/978-3-032-37627-5_13. 2

Ziyi Wang, Siva Rajesh Kasa, Ankith M S, Santhosh Kumar Kasa, Jiaru Zou, Sumit Negi, Ruqi Zhang, Nan Jiang, and Qifan Song. DIVERSED: Relaxed speculative decoding via dynamic ensemble verification. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 2341–2349. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/wang26d.html. 2

Yepeng Weng, Qiao Hu, Xujie Chen, Li Liu, Dianwen Mei, Huishi Qiu, Jiang Tian, and Zhongchao Shi. Traversal verification for speculative tree decoding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 77899–77924. Curran Associates, Inc., 2025. doi: 10.52202/085713-2611. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/70596d70542c51c8d9b4e423f4bf2736-Paper-Conference.pdf. 2

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115. 8

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf. 8

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan,

and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32011–32050, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8766fbc68e1ed1cdef712ce273e0a363-Paper-Conference.pdf. 2

Yuxuan Zhou, Fei Huang, Heng Li, Fengyi Wu, Tianyu Wang, Jianwei Zhang, Junyang Lin, and Zhi-Qi Cheng. Overcoming joint intractability with lossless hierarchical speculative decoding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 108413–108446, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b06dbe97e29ba06bdfdb6949191d2591-Paper-Conference.pdf. 1, 2, 3

Xiandong Zou, Jianshu Li, Jing Huang, and Pan Zhou. Variational speculative decoding: Rethinking draft training from token likelihood to sequence acceptance. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=7xO0ADKJu8. 2

A Proofs for the Retained-Mass Objective

A.1 BV–HSD Equivalence

We first prove the state identity in Theorem 1 and then extend it to the complete verification rule. All ratios are evaluated on realized prefixes with positive proposal probability. For notational simplicity, the branch-wise derivation assumes full-support token distributions, as in softmax language models; the equivalent mass identities extend to zero-probability branches by continuity.

Lemma 8 (Running-Maximum Form)**.** *Let*

$$R_t := \frac{p(X_{1:t})}{r(X_{1:t})}, \qquad M_t := \max_{0\le j\le t} R_j, \qquad R_0 = 1. \tag{30}$$

Then the BV state and the HSD capped prefix ratio satisfy

$$w_t = \frac{R_t}{M_t}, \qquad R_t^\star = \frac{R_t}{M_{t-1}}. \tag{31}$$

Proof. Let $\lambda_t = p(X_t \mid X_{<t})/r(X_t \mid X_{<t})$, so that $R_t = R_{t-1}\lambda_t$. Expanding the BV recursion gives

$$\begin{aligned} w_t &= \min\{1, \lambda_t, \lambda_{t-1}\lambda_t, \dots, \lambda_1 \cdots \lambda_t\} \\ &= \min_{0\le j\le t} \frac{R_t}{R_j} = \frac{R_t}{M_t}. \end{aligned} \tag{32}$$

In capped-ratio HSD, the reference prefix is a preceding prefix attaining M_{t-1}. The likelihood ratio of the suffix after that prefix is therefore $R_t/M_{t-1} = R_t^\star$. □

Proof of Theorem 1. By Lemma 8,

$$R_t^\star = \frac{R_t}{M_{t-1}} = \frac{R_{t-1}}{M_{t-1}} \frac{p(X_t \mid X_{<t})}{r(X_t \mid X_{<t})} = w_{t-1} \frac{p(X_t \mid X_{<t})}{r(X_t \mid X_{<t})}. \tag{33}$$

Truncating at one gives the BV update, so $\min\{1, R_t^\star\} = w_t$.

It remains to compare the actual acceptance and correction rules. Fix a proper prefix $s = X_{1:t}$, and write $w = w_t$, $p_x = p(x \mid s)$, and $r_x = r(x \mid s)$. Applying Eq. (33) to child sx gives

$$R^\star(sx) = w \frac{p_x}{r_x}. \tag{34}$$

Define

$$G_t := \sum_x [wp_x - r_x]_+. \tag{35}$$

The proper-prefix acceptance probability of BV is

$$h_t^{\mathrm{BV}} = \frac{G_t}{G_t + 1 - w}. \tag{36}$$

The positive and negative capped branch divergences of HSD become

$$D^\star_{+,t} = r(s) \sum_x [wp_x - r_x]_+ = r(s) G_t, \tag{37}$$

$$D^\star_{-,t} = r(s) \sum_x [r_x - wp_x]_+. \tag{38}$$

Using $[a]_+ - [-a]_+ = a$ and $\sum_x r_x = \sum_x p_x = 1$,

$$\sum_x [r_x - wp_x]_+ = G_t + 1 - w. \tag{39}$$

Hence

$$h_t^{\mathrm{HSD}} = \frac{D^\star_{+,t}}{D^\star_{-,t}} = \frac{G_t}{G_t + 1 - w} = h_t^{\mathrm{BV}}. \tag{40}$$

When a proper prefix is selected, BV samples the correction token from

$$p_{\mathrm{res}}^{\mathrm{BV}}(x \mid s) = \frac{[wp_x - r_x]_+}{\sum_y [wp_y - r_y]_+}. \tag{41}$$

The HSD branch residual is proportional to

$$r(sx)[R^\star(sx) - 1]_+ = r(s)[wp_x - r_x]_+, \tag{42}$$

so the common factor $r(s)$ cancels and the two correction distributions coincide. For a complete block, both capped-ratio HSD and BV accept with probability $\min\{1, R_L^\star\} = w_L$. Finally, BV's forward scan and HSD's backward scan select the largest successful prefix index from the same independent prefix tests. Coupling them with the same uniforms gives the same accepted length and output almost surely. □

A.2 Note on the Released HSD Full-Block Rule

The equivalence above concerns the capped-ratio verifier analyzed by HSD. Its complete-block acceptance probability is

$$\min\{1, R_L^\star\} = \min\left\{1, \frac{R_L}{M_{L-1}}\right\} = w_L. \tag{43}$$

In the released implementation used in the prior empirical comparison, the complete-block branch instead uses the ordinary joint ratio $\min\{1, R_L\}$. These quantities differ whenever an earlier prefix ratio exceeds one, so the released rule can report a larger full-block acceptance probability than the theoretical capped verifier. Our equivalence claim refers to the theoretical formulation; empirical comparisons must use the same verification law and systems configuration.

Table 3: Empirical comparison of BV, theoretical HSD, and the released HSD implementation on 500 GSM8K examples using a single NVIDIA RTX 4090.

Method	N	Acc.	BE	E2E	Full	Peak GiB
BV	500	0.0880	5.5699	40.237	0.4153	15.758
HSD (Theory)	500	0.0880	5.5699	40.352	0.4153	15.758
HSD	500	0.0900	5.7063	41.282	0.4513	15.758

We further verify this distinction empirically on GSM8K. As shown in Table 3, HSD (Theory), implemented according to the theoretical capped-ratio rule, exactly matches BV in block efficiency, full-block acceptance rate, and task accuracy. Their small difference in end-to-end throughput is attributable to implementation-level runtime variation, since BV and HSD (Theory) realize the same verification law through different code paths. In contrast, HSD, which denotes the released implementation, produces different acceptance statistics, consistent with the distinct full-block rule described above.

A.3 Retained-Mass Identities

We prove Proposition 2. Multiplying the BV update in Eq. (2) by $r(sx) = r(s)r(x \mid s)$ gives

$$\begin{aligned}\mu_r(sx) &= r(sx)\min\left\{1, w_r(s)\frac{p(x \mid s)}{r(x \mid s)}\right\} \\ &= \min\{r(sx), r(s)w_r(s)p(x \mid s)\} \\ &= \min\{r(sx), \mu_r(s)p(x \mid s)\},\end{aligned} \tag{44}$$

which proves Eq. (6).

For a prefix s of depth t, retained mass is the joint probability that the proposal follows s and verification retains at least t draft tokens:

$$\mu_r(s) = \Pr(X_{1:t} = s, \tau \geq t). \tag{45}$$

The events for different depth-t prefixes are disjoint, hence

$$\sum_{s\in\mathcal{V}^t} \mu_r(s) = \Pr(\tau \geq t). \tag{46}$$

Applying the tail-sum identity for $\tau \in \{0, \ldots, L\}$,

$$\begin{aligned} J(r) &= 1 + \sum_{t=1}^{L} \sum_{s\in\mathcal{V}^t} \mu_r(s) \\ &= 1 + \sum_{t=1}^{L} \Pr(\tau \geq t) = 1 + \mathbb{E}[\tau] = \mathrm{BE}(r), \end{aligned} \tag{47}$$

which proves Eq. (7).

Related retained-mass identities and the corresponding multi-path block-efficiency objective also appear in Thomas & Pal (2026) (Appendix C and Eq. (21)). Here, we use the local supply–capacity recursion to support the realizable proposal construction and retained-flow optimization developed in Sections 5–6.

B Proofs for Sequential Martingale Selection

B.1 Finite-K Selector Constraints

An antichain $T \subseteq \mathcal{V}^{\leq L}$ contains no two prefixes for which one is an ancestor of the other. The finite-K characterization of GBV (Thomas & Pal, 2026) states that a complete-path distribution r belongs to $\mathcal{R}_K(q)$ if and only if every antichain satisfies

$$\sum_{s\in T} r(s) \leq 1 - \left(1 - \sum_{s\in T} q(s)\right)^K. \tag{48}$$

The right-hand side is the probability that at least one of the K draft paths reaches the antichain. The singleton choice $T = \{s\}$ yields

$$r(s) \leq 1 - (1 - q(s))^K \leq Kq(s), \qquad 0 \leq \frac{r(s)}{q(s)} \leq K. \tag{49}$$

This density bound is sufficient for the universal coverage result.

B.2 Proof of Prefix-Density Conservation

For $A_s = r(s)/q(s)$, consistency of prefix marginals gives

$$\begin{aligned} A_s &= \frac{1}{q(s)} \sum_x r(sx) = \frac{1}{q(s)} \sum_x q(s)q(x \mid s)A_{sx} \\ &= \sum_x q(x \mid s)A_{sx}, \end{aligned} \tag{50}$$

which proves Lemma 3. Equivalently,

$$A_s = \mathbb{E}_q\left[\frac{r(X)}{q(X)} \,\middle|\, X \succeq s\right], \tag{51}$$

so the prefix density is the complete-path density conditioned on the information available at s.

B.3 Proof of Exact Finite-K Realization

We prove Theorem 4. Let $Y^{(1)}, \ldots, Y^{(K)} \overset{\text{i.i.d.}}{\sim} q$ denote the randomly ordered candidates. Since $\{\alpha_s\}$ is a bounded q-martingale with root value m,

$$\mathbb{E}_{X\sim q}[\alpha_X] = m, \qquad \mathbb{E}_{X\sim q}[1 - \alpha_X] = 1 - m. \tag{52}$$

Fix a complete path X. If it appears at position $i < K$, it is selected when the first $i-1$ candidates reject and its own gate accepts, an event of probability

$$q(X)(1-m)^{i-1}\alpha_X. \tag{53}$$

If it appears in the final position, it is selected after all preceding gates reject, with probability $q(X)(1-m)^{K-1}$. Summing over the mutually exclusive positions gives

$$\begin{aligned} q^\Gamma(X) &= q(X)\left[\alpha_X \sum_{i=0}^{K-2}(1-m)^i + (1-m)^{K-1}\right] \\ &= q(X)\left[C_m\alpha_X + d_m\right] = q(X)A_X. \end{aligned} \tag{54}$$

Because $A_s = d_m + C_m\alpha_s$ is itself a q-martingale,

$$q^\Gamma(s) = \sum_{X \succeq s} q(X)A_X = q(s)\mathbb{E}_q[A_X \mid X \succeq s] = q(s)A_s \tag{55}$$

Consequently, it gives

$$q^\Gamma(x \mid s) = q(x \mid s)\frac{A_{sx}}{A_s}. \tag{56}$$

Thus every proposal in $\mathcal{S}_{K,m}(q)$ has the claimed explicit finite-K realization.

B.4 Proof of Radial and Objective Coverage

We prove Theorem 5 in three steps.

Radial coverage. Fix $r \in \mathcal{R}_K(q)$ and let $A_s^r = r(s)/q(s)$. By Eq. (49), $0 \le A_s^r \le K$. For $m = 1/K$, define

$$\alpha_s := \frac{A_s^r}{K}. \tag{57}$$

This is a valid bounded gate with root value $1/K$. Since $C_K = K(1-d_K)$, its SMS density is

$$\bar{A}_s = d_K + C_K\alpha_s = d_K + (1-d_K)\frac{r(s)}{q(s)}. \tag{58}$$

Multiplying by $q(s)$ gives

$$\bar{r} = d_K q + (1-d_K)r \in \mathcal{S}_K(q). \tag{59}$$

Since r was arbitrary and $\mathcal{S}_K(q) \subseteq \mathcal{R}_K(q)$ by Theorem 4, Eq. (17) follows.

Concavity of retained mass. For a prefix $s_{1:i}$, repeatedly applying Eq. (6) yields

$$\mu_r(s_{1:i}) = \min_{0 \le k \le i}\left\{r(s_{1:k})p(s_{k+1:i} \mid s_{1:k})\right\}, \tag{60}$$

where $r(s_{1:0}) = 1$. Each term inside the minimum is affine in r, so each prefix retained mass, and hence $J(r)$, is concave:

$$J((1-\lambda)r_0 + \lambda r_1) \ge (1-\lambda)J(r_0) + \lambda J(r_1). \tag{61}$$

Objective coverage. Let $r^\star$ maximize J over $\mathcal{R}_K(q)$. The radial point $\bar{r}^\star = d_K q + (1-d_K)r^\star$ belongs to $\mathcal{S}_K(q)$, and concavity gives

$$J_{\mathcal{S}}^\star \ge J(\bar{r}^\star) \ge d_K J(q) + (1-d_K)J_{\mathcal{R}}^\star. \tag{62}$$

Rearranging proves Eq. (18).

Why $m = 1/K$. For any $m \le 1/K$, the same argument uses $\alpha_s = mA_s^r$ and yields radial coefficient

$$\lambda_K(m) = 1 - (1-m)^{K-1}. \tag{63}$$

The restriction $m \le 1/K$ ensures $\alpha_s \le 1$ for every realizable density. Since $\lambda_K'(m) = (K-1)(1-m)^{K-2} > 0$, the largest universal coefficient is attained at $m = 1/K$.

C Proofs and Details for Retained-Flow Optimization

C.1 Full-Horizon SMS Optimization

The practical method optimizes only the next layer. Here we state the exact full-horizon problem that it approximates. For a positive-probability prefix s, write $p_x = p(x \mid s)$ and $q_x = q(x \mid s)$. For normalized states (A, B), let

$$V_s(A, B) := \max_{\pi} \frac{1}{q(s)} \sum_{\substack{u:\, s \prec u \\ |u| \le L}} \mu_\pi(u), \tag{64}$$

where π ranges over feasible SMS flow policies below s. Set $V_s(A, B) = 0$ at depth L, and for $|s| < L$ define

$$g_x(z; B) := \min\{z_x, Bp_x\}. \tag{65}$$

Proposition 9 (Retained-Mass Bellman Equation)**.** *The exact value satisfies*

$$V_s(A, B) = \max_{z \in \mathcal{Z}_s(A)} \sum_{x: q_x > 0} \left[g_x(z; B) + q_x V_{sx}\left(\frac{z_x}{q_x}, \frac{g_x(z; B)}{q_x} \right) \right]. \tag{66}$$

Consequently,

$$J^\star_{\mathcal{S}} = 1 + V_\varnothing(1, 1). \tag{67}$$

Proof. For a fixed first-layer flow z, the retained mass at child sx is $q(s)g_x(z; B)$, while the best strict-descendant contribution below that child is $q(s)q_x V_{sx}(z_x/q_x, g_x(z; B)/q_x)$. Once the child states are fixed, the subtrees are independent under the prefix-local SMS constraints. Summing over children and maximizing over z proves Eq. (66); adding the root mass proves Eq. (67). □

The first term in Eq. (66) is exactly LP1. Dropping the continuation term gives the horizon-one truncation used by SMS-LP1: it is exact for the next layer, while target-dependent future value is deliberately omitted to avoid additional target-model evaluation.

C.2 LP1 as a Flow Projection

Recall $c_x = B_s p_x$, $\sum_x z_x = A_s$, and $\sum_x c_x = B_s$. For nonnegative a, b,

$$\min\{a, b\} = \frac{a + b - |a - b|}{2}. \tag{68}$$

Applying this identity coordinate-wise gives

$$\sum_x \min\{z_x, c_x\} = \frac{A_s + B_s - \|z - c\|_1}{2}. \tag{69}$$

Hence the LP1 solution set is

$$\mathcal{Z}^\star_s = \arg \min_{z \in \mathcal{Z}_s(A_s)} \|z - c\|_1. \tag{70}$$

This proves the projection interpretation used in Sec. 6.2.

C.3 Proof of the Minimum-Movement Completion

We prove Theorem 6. Let

$$F(z) := \sum_x \min\{z_x, c_x\} \tag{71}$$

be the LP1 objective and let $z^0 = A_s q$ be the unmodified draft flow. Define a^-_x, a^+_x, D^-, D^+, and Δ as in Eqs. (27)–(28).

Feasibility. If $D^- > 0$, the donor fraction Δ/D^- lies in $[0, 1]$; otherwise $\Delta = 0$ and the donor term vanishes. The same holds for the receiver fraction. Hence donor coordinates remain above $\max\{\ell_x, c_x\} \geq \ell_x$, receiver coordinates remain below $\min\{u_x, c_x\} \leq u_x$, and all other coordinates remain at z_x^0. Moreover,

$$\sum_x z_x^{\mathrm{MM}} = \sum_x z_x^0 - \Delta + \Delta = A_s, \tag{72}$$

so $z^{\mathrm{MM}} \in \mathcal{Z}_s(A_s)$.

LP1 optimality. Removing up to a_x^- from a donor does not reduce $\min\{z_x, c_x\}$, whereas adding up to a_x^+ to a receiver increases the objective one-for-one. Therefore

$$F(z^{\mathrm{MM}}) - F(z^0) = \Delta. \tag{73}$$

No feasible reallocation can improve by more than D^+, the total useful receiver room. Nor can it improve by more than D^-: after all removable overflow is exhausted, each additional unit removed loses one unit of retained mass and cancels the corresponding receiver gain. Thus every feasible z satisfies

$$F(z) - F(z^0) \leq \min\{D^-, D^+\} = \Delta, \tag{74}$$

proving that z^{MM} solves LP1.

Minimum movement. For any feasible z, mass conservation implies

$$\sum_x [z_x - z_x^0]_+ = \sum_x [z_x^0 - z_x]_+ = \frac{1}{2}\|z - z^0\|_1. \tag{75}$$

An LP1 optimum must improve F over z^0 by Δ, while each unit of positive mass movement improves F by at most one. Hence every optimum has $\|z - z^0\|_1 \geq 2\Delta$. The proposed allocation moves exactly Δ out of donors and Δ into receivers, so

$$\|z^{\mathrm{MM}} - z^0\|_1 = 2\Delta. \tag{76}$$

It therefore minimizes movement among all LP1 optima. Proportional allocation within the donor and receiver sets makes the returned optimizer deterministic and permutation equivariant.

C.4 Proof of SMS-LP1 Losslessness

We prove Theorem 7. The case $K = 1$ is ordinary BV, so assume $K \geq 2$. At every visited prefix, the minimum-movement solver returns a deterministic flow $z_s \in \mathcal{Z}_s(A_s)$. Hence

$$\sum_x z_s(x) = A_s, \qquad d_K q(x \mid s) \leq z_s(x) \leq (d_K + C_K) q(x \mid s). \tag{77}$$

For $q(x \mid s) > 0$, define

$$A_{sx} := \frac{z_s(x)}{q(x \mid s)}, \qquad \alpha_{sx} := \frac{A_{sx} - d_K}{C_K}. \tag{78}$$

The box constraints imply $0 \leq \alpha_{sx} \leq 1$, and flow conservation gives

$$\sum_x q(x \mid s)\alpha_{sx} = \frac{\sum_x z_s(x) - d_K}{C_K} = \frac{A_s - d_K}{C_K} = \alpha_s. \tag{79}$$

At the root, $A_\varnothing = 1 = d_K + C_K/K$, so $\alpha_\varnothing = 1/K$. The generated gates therefore form a valid bounded martingale process.

The minimum-movement rule depends only on the prefix state and the local $p(\cdot \mid s)$ and $q(\cdot \mid s)$. Identical prefixes consequently receive identical gates, independently of the other candidates in the current sample. Thus the candidate-set-independence required by Theorem 4 holds, and that theorem gives

$$q^\Gamma(s) = q(s)A_s, \qquad q^\Gamma(x \mid s) = q(x \mid s)\frac{A_{sx}}{A_s} = \frac{z_s(x)}{A_s}. \tag{80}$$

Algorithm 1 stores precisely these conditionals and passes them to BV after sequential selection. Since BV is lossless for every valid proposal, the final output distribution is exactly p.

D Implementation and Experimental Details

D.1 Experimental Setup and Measurement

Models and benchmarks. We evaluate Qwen2.5-0.5B-Instruct with Qwen2.5-7B/14B/32B-Instruct, Llama-3.2-1B-Instruct with Llama-3.1-8B-Instruct, and Muse-Glimmer-30B-DFlash2 with Muse-Glimmer-30B. We use 1,319 GSM8K, 500 MATH-500, 164 HumanEval, and 257 MBPP-sanitized examples, as well as 80 two-turn MT-Bench questions. Each method's first-turn response provides its own second-turn context.

Comparison protocol. All methods use the same models, prompts, and target sampling settings within each draft–target pair, on an NVIDIA RTX PRO 6000 GPU with batch size one. Tokenwise and BV use one draft path; GBV and SMS-LP1 each use K paths from the same draft proposal. Main results use $K = 6$, $L = 8$, $T_p = T_q = 1$. Thus, main-table comparisons with GBV use matched K; Table 2 separately reports the best observed K on one Qwen2.5-32B GSM8K setting. We use an 8-token warm-up for each method. BE is computed as the total number of tokens produced by verification divided by the number of target verification rounds; scoring K paths in one batch counts as one round. E2E is computed as pooled generated tokens divided by pooled wall-clock time across prompts, including prefill and decoding, with component profiling disabled. In the DFlash2 runs, the initial token produced during prefill is excluded from BE.

D.2 Draft Proposals and Verifier Implementation

Autoregressive drafting. For the Qwen and Llama model pairs, we sample K independent paths from the drafter's deployed sampling distribution q, recording its conditional probabilities along each path. The target scores the candidate paths in a batched forward pass before GBV or SMS-LP1 selects one. Both methods use the resulting target distributions and the same BV, correction, and cache-commit implementation. SMS-LP1 computes its local flow updates from these already available target distributions; it does not require an additional target forward pass beyond candidate scoring.

DFlash2 drafting and proposal probabilities. We extend the official DFlash2 implementation to sample multiple paths without retraining or replacing its learned candidate selector. Given the current decoding context, one draft forward pass produces shared hidden states and a candidate set C_t at each position t. We generate K paths using independent categorical draws from these conditionals and record the probabilities used at each step. Thus, given the decoding context, the paths are i.i.d. draws from the same autoregressive proposal q. DFlash2's learned selector samples tokens within each path; GBV and SMS-LP1 subsequently select among the complete paths.

Target scoring and verification. All sampled paths remain in the selection pool, including duplicates. Identical complete paths are scored by the target once, and their logits are mapped back to every sampled occurrence. Target scoring therefore precedes path selection. Selection uses target probabilities on the draft support, while the selected branch's full-vocabulary target distribution is used for BV and its correction or bonus sampling. The recorded q is used to compute the induced selected-path proposal passed to BV.

E Additional Results and Analyses

E.1 Stability Across Random Seeds

We assess seed stability using three random seeds. Tables 4 and 5 report dataset-level means and standard deviations. The Qwen comparison uses GBV as its baseline, while the DFlash2 comparison uses tokenwise verification, as specified in the respective captions.

As shown in Tables 4 and 5, SMS-LP1 improves both BE and E2E across all five datasets and three seeds. On Qwen, its mean paired gains over GBV reach up to 12.29% in BE and 14.57% in E2E.

The DFlash2 results show that the improvement is present in both verification efficiency and measured throughput. SMS-LP1 exceeds tokenwise verification, BV, and GBV on both metrics in all 15 dataset–

seed comparisons. Its paired E2E gain over tokenwise verification, averaged equally across datasets within each seed, remains between 5.67% and 6.56% across the three seeds; the dataset-level mean gains reach up to 20.70% in BE and 11.25% in E2E. The consistent BE gains show an improvement in generated tokens per target evaluation, while the corresponding E2E gains show that this benefit persists after full decoding overhead. Together, these results support a reproducible performance improvement in the evaluated DFlash2 setting, rather than an isolated favorable timing run.

Table 4: Qwen2.5 seed stability. Results for the Qwen2.5-0.5B/32B pair ($K = 6$, $L = 8$, $T = 1$). Values are mean $\pm$ sample standard deviation across three seeds; E2E is measured in generated tokens per second. Gain is computed relative to GBV for each paired seed before aggregation.

Dataset	N	Metric	GBV	SMS-LP1	Gain (%)
GSM8K	1319	BE	6.10 ± 0.04	6.62 ± 0.03	8.59 ± 1.09
		E2E	31.98 ± 0.32	35.22 ± 0.36	10.13 ± 2.02
MATH-500	500	BE	6.26 ± 0.05	6.95 ± 0.02	10.98 ± 0.72
		E2E	34.58 ± 1.54	39.37 ± 1.54	13.87 ± 0.81
HumanEval	164	BE	5.89 ± 0.06	6.48 ± 0.05	10.02 ± 1.90
		E2E	33.66 ± 0.48	37.89 ± 0.10	12.60 ± 1.86
MBPP	257	BE	5.64 ± 0.09	6.33 ± 0.07	12.29 ± 2.53
		E2E	32.36 ± 0.44	37.07 ± 0.79	14.57 ± 3.06
MT-Bench	80	BE	4.09 ± 0.02	4.48 ± 0.03	9.49 ± 0.74
		E2E	23.20 ± 0.16	26.06 ± 0.33	12.31 ± 0.81

Table 5: DFlash2 seed stability. Results for the DFlash2/Glimmer-30B pair ($K = 6$, $L = 8$, $T = 1$). Values are mean $\pm$ sample standard deviation across three seeds; E2E is measured in generated tokens per second. SMS-LP1 gain is computed relative to tokenwise verification for each paired seed before aggregation.

Dataset	N	Metric	Tokenwise	BV	GBV	SMS-LP1	Gain (%)
GSM8K	1319	BE	6.753 ± 0.029	6.833 ± 0.022	7.154 ± 0.028	7.367 ± 0.014	9.09 ± 0.52
		E2E	107.74 ± 0.34	108.84 ± 0.64	108.48 ± 0.60	111.24 ± 0.18	3.25 ± 0.48
MATH-500	500	BE	5.995 ± 0.004	6.072 ± 0.015	6.456 ± 0.013	6.763 ± 0.021	12.80 ± 0.41
		E2E	95.62 ± 0.47	96.65 ± 0.41	96.94 ± 0.56	101.02 ± 0.28	5.64 ± 0.47
HumanEval	164	BE	7.101 ± 0.103	7.148 ± 0.111	7.290 ± 0.080	7.629 ± 0.066	7.45 ± 1.99
		E2E	109.86 ± 2.81	110.42 ± 3.39	107.97 ± 2.27	112.70 ± 2.73	2.60 ± 1.96
MBPP	257	BE	4.721 ± 0.039	4.858 ± 0.027	5.007 ± 0.043	5.698 ± 0.037	20.70 ± 0.56
		E2E	76.43 ± 1.14	78.36 ± 0.49	75.18 ± 0.32	85.02 ± 0.85	11.25 ± 0.76
MT-Bench	80	BE	4.516 ± 0.121	4.645 ± 0.026	4.908 ± 0.018	5.336 ± 0.092	18.19 ± 2.01
		E2E	72.68 ± 1.82	74.52 ± 0.37	72.64 ± 0.29	78.51 ± 1.38	8.05 ± 1.65

E.2 Runtime Breakdown

We profile BV, GBV, and SMS-LP1 on 200 GSM8K prompts with the Qwen2.5-0.5B/32B draft–target pair. We use $L = 8$, $T = 1$, and $K = 6$ for GBV and SMS-LP1; BV uses one draft path. Table 6 reports the accumulated time of each component divided by the number of generated tokens. CUDA is synchronized at component boundaries for this analysis; the end-to-end throughput results in the main text are measured separately without this profiling instrumentation.

Table 6: Runtime breakdown on 200 GSM8K prompts. Times are milliseconds per generated token; percentages are within each method.

	GBV		SMS-LP1	
Component	ms/token	%	ms/token	%
Total	32.142	100.00	29.573	100.00
Prefill	1.660	5.16	1.651	5.58
Draft forward	13.853	43.10	12.871	43.52
Target forward	12.953	40.30	12.014	40.63
Logits processing	0.268	0.83	0.250	0.85
Selector	0.937	2.92	0.490	1.66
Verification	0.111	0.35	0.107	0.36
Cache / commit	0.555	1.73	0.514	1.74
Other	1.804	5.61	1.675	5.66

Compared with GBV, SMS-LP1 reduces profiled time from 32.142 to 29.573 ms per generated token. Its selector accounts for 1.66% of total time, compared with 2.92% for GBV. The lower draft and target forward times per generated token are consistent with SMS-LP1's higher block efficiency.

E.3 Comparison with Multi-Path Verifiers

We further examine SpecTr-GBV (Lin et al., 2026), another recent multi-path block-verification method.

A Losslessness Issue in SpecTr-GBV Consider $\mathcal{V} = \{A, B\}$, $K = 2$, and $L = 2$. In our notation, q is the draft distribution and p is the target distribution. Specify their two-token block distributions by

x	AA	AB	BA	BB
$q(x)$	$1/6$	$1/2$	$1/6$	$1/6$
$p(x)$	$1/4$	$1/4$	$1/6$	$1/3$

and draw the two draft blocks independently from q. In particular, $p(A) = 1/2$. Substitution into Eqs. (2)–(4) of SpecTr-GBV gives prefix acceptance probabilities $h_1(A) = 1/15$ and $h_1(B) = 1/5$, full-block acceptance probabilities $(8/11, 1/3, 6/11, 9/11)$ in the order (AA, AB, BA, BB), and zero root residual probability for A. Enumerating the 16 ordered draft-block pairs under its Algorithm 1, gives

$$\Pr(Y_1 = A) = \frac{133}{363} + \frac{67}{3630} = \frac{127}{330} \neq p(A) = \frac{1}{2}.$$

Thus the published rule, interpreted literally, does not preserve the target distribution on this full-support instance.