ATTAINING AND SURPASSING THE IID CEILING IN MULTI-DRAFT SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-draft speculative decoding has a computable acceptance ceiling under independent and identically distributed (IID) proposals, but turning this value into an exact, scalable sampler remains a constructive challenge. We resolve this problem with *Optimal IID*, which preserves the target distribution exactly and attains the IID ceiling for any candidate count without iterative optimization or an approximation tolerance. Its finite-score construction uses near-linear arithmetic preprocessing. To exceed this ceiling, *Stratified* changes proposal dependence through equal-mass stratification, preserving each slot's draft marginal and matching or exceeding IID coverage for every token subset. Its optimal acceptance is no lower than the IID ceiling for any target, with strict improvements on some inputs. A recursive exact verifier attains this new ceiling with near-linear arithmetic work and polylogarithmic arithmetic span. Across three model pairs from the Qwen3 and Gemma 2 families, Stratified recovers approximately 85–92% of the gap from optimal IID to a target-aware coupling oracle with the same proposal marginals. The evaluation uses four candidates, top-100 draft support, and temperature 1. In a feasibility study at this setting, a single-GH200 implementation for Qwen3-32B/1.7B averages 0.49 ms per output token for verification-plan construction and online sampling, including residual correction, and accelerates decoding by 1.58× over autoregressive sampling. Proposal dependence thus enables further acceptance gains after IID verification is optimal.

1 INTRODUCTION

Speculative decoding accelerates a target language model by verifying proposals from a cheaper draft model while preserving the target distribution (Leviathan et al., 2023; Chen et al., 2023). Multi-draft methods evaluate several candidates in parallel; accepting one allows decoding to continue along an already evaluated branch (Sun et al., 2023). For a given draft model and candidate count, what is the largest acceptance probability compatible with the target distribution, and how can a verifier attain it?

The optimal verifier solves a coupling problem whose direct formulation grows exponentially with the candidate count (Sun et al., 2023). For independent and identically distributed (IID) proposals, canonical decomposition exposes a selector-and-correction structure (Khisti et al., 2025); subset characterizations make the optimal acceptance efficiently computable and quantify the gap left by practical verifiers (Hu et al., 2025). Computing this value does not by itself yield an optimal sampler. Global Resolution reduces the construction to convex optimization in at most one variable per vocabulary token, with tunable bounds on target-marginal and acceptance error (Thomas & Pal, 2026). These advances leave a precise constructive question: can the IID ceiling be attained exactly by an explicit, scalable sampling procedure?

We resolve this constructive problem with *Optimal IID*. For any number of IID candidates, it exactly preserves the target distribution and exactly attains the optimal candidate-membership probability, without iterative optimization or an approximation tolerance. The finite-score construction has near-linear arithmetic preprocessing and linear storage; we demonstrate it at full LLM vocabulary scale. The IID ceiling thus becomes an executable sampling primitive.

Once this ceiling is attained, no verifier can further increase acceptance under the same IID proposal law. Hu et al. (2025) also pursue acceptance gains through non-IID proposal design. However, even their greedy law's optimal acceptance can fall below the IID ceiling: Thomas & Pal (2026, Appendix

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.

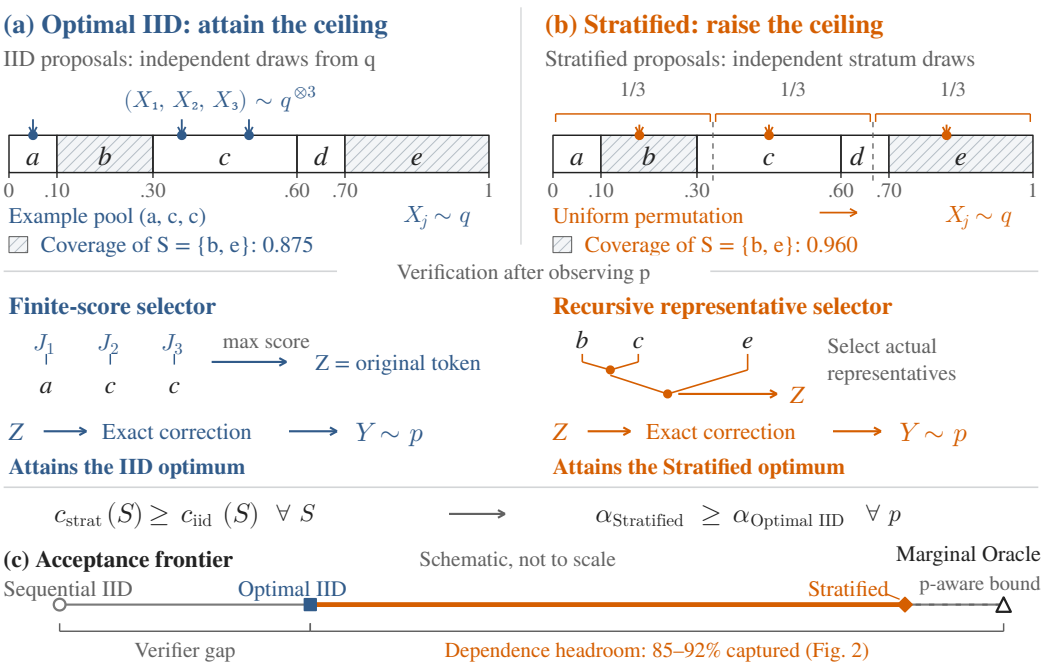

Figure 1: **Attaining and surpassing the IID ceiling with exact samplers.** (a,b) For $q = (.10, .20, .30, .10, .30)$ and $n = 3$, stratification increases the coverage probability of the hatched subset while preserving each slot's marginal. Pools are illustrative; repeats remain possible and source labels are retained. The finite-score and recursive selectors each attain their proposal law's optimum after correction. (c) Orange marks recovered dependence headroom; the measured fraction is 85–92% across three model pairs at $n = 4$, $K = 100$, $T = 1$ (Figure 2).

O) report lower acceptance at all tested draft supports $K \geq 100$ on Gemma-2 and Llama-3. We seek a stronger guarantee: surpass optimal IID through *dependence alone*, keeping the candidate count and every slot's draft marginal q unchanged, with no loss for any target distribution.

Our *Stratified* construction achieves this goal by coordinating candidates across equal portions of draft probability mass (Figure 1). Formed before observing the current target distribution, the proposal law preserves every slot's marginal q and covers every token subset with probability at least as high as IID. Its optimal acceptance is therefore no lower than the IID ceiling for every target, with strict improvements on some inputs. An exact recursive verifier attains this new fixed-law optimum, turning the coverage advantage into an executable sampler.

We quantify the remaining dependence headroom through the *Marginal Oracle*, the optimum over arbitrary proposal couplings that preserve every slot's marginal and may use the target distribution. Stratified recovers approximately 85–92% of the gap from optimal IID to this reference across three Qwen3 and Gemma 2 model pairs, using four candidates, top-100 draft support, and temperature 1.

Our contributions.

- **Exact IID attainment.** An explicit optimal sampler for arbitrary candidate count, with near-linear arithmetic preprocessing and linear storage (Theorem 1).
- **A constructive guarantee beyond IID.** Same-marginal coverage dominance over IID and an exact optimal Stratified verifier with near-linear arithmetic work and polylogarithmic span (Theorems 2–3).
- **Quantifying dependence headroom.** The target-aware same-marginal optimum (Proposition 4), matched-state frontier comparisons (Figure 2), and end-to-end feasibility evidence.

2 EXACT VERIFICATION AS A COVERAGE PROBLEM

At one decoding position, let $p, q \in \Delta_V$ be the target and draft distributions on $E = [V]$. Let $n \geq 1$ be the number of proposal slots, with repeated tokens allowed. The proposal stage generates $X = (X_1, \ldots, X_n)$ from a law $\mathcal{Q}$ determined before the current p is available. After observing p and the candidates, a randomized verifier outputs Y. Exactness means $Y \sim p$, integrating over both proposal and verifier randomness. We measure *candidate membership*: $\alpha = \mathbb{P}(Y \in C(X))$, where $C(X) = \{X_1, \ldots, X_n\}$. Thus each token is counted once even when proposed repeatedly.

For a fixed law $\mathcal{Q}$, define its coverage function and optimal acceptance by

$$c_{\mathcal{Q}}(S) = \mathbb{P}(C(X) \cap S \neq \varnothing), \qquad \alpha^*(p, \mathcal{Q}) = \max_{X \sim \mathcal{Q},\, Y \sim p} \mathbb{P}(Y \in C(X)). \tag{1}$$

The coverage function is normalized, monotone, and submodular. The established max-flow/min-cut characterization (Hu et al., 2025; Thomas & Pal, 2026) gives

$$\alpha^*(p, \mathcal{Q}) = 1 - D_{\mathcal{Q}}(p), \qquad D_{\mathcal{Q}}(p) = \max_{S \subseteq E} \{p(S) - c_{\mathcal{Q}}(S)\}. \tag{2}$$

For any subset S, success inside S requires the proposal pool to hit S; hence excess target mass $p(S) - c_{\mathcal{Q}}(S)$ must be rejected. A subset attaining the largest deficit is an acceptance *bottleneck*. Max-flow shows that this bottleneck determines the optimum, separating the coverage supplied by a proposal law from the verifier that realizes it.

Our constructions use the selector-and-correction structure of Khisti et al. (2025). A selector returns an actual candidate $Z \in C(X)$ with marginal ρ. Its feasible marginals form the coverage base

$$\mathcal{B}(c) = \{\rho \geq 0 : \rho(E) = 1,\ \rho(S) \leq c(S) \text{ for all } S \subseteq E\}. \tag{3}$$

If $\mathrm{TV}(p, \rho) = D_{\mathcal{Q}}(p)$, keep Z with probability $\min(1, p_Z/\rho_Z)$ and otherwise draw from $(p - \rho)_+ / D_{\mathcal{Q}}(p)$. The output has distribution p, and the probability of retaining Z is $1 - D_{\mathcal{Q}}(p)$. Since retained outputs belong to the pool and Equation (2) bounds the total membership probability, the complete algorithm attains $\alpha^*(p, \mathcal{Q})$. Appendix A gives the flow argument, including zero-probability cases.

3 ATTAINING THE IID CEILING

The IID ceiling is computable; the remaining task is to realize it by sampling. Our construction has three stages: (1) choose an optimal candidate marginal ρ; (2) attach an independent finite-valued random score to each candidate so that the highest-scoring position carries a token with law ρ; (3) correct that token to p.

Under IID proposals, $c_{\mathrm{iid}}(S) = 1 - (1 - q(S))^n$. Sort the m positive-q tokens by nondecreasing p_i/q_i and set $Q_j = \sum_{i \leq j} q_i$, $P_j = \sum_{i \leq j} p_i$, including the empty prefix. The known prefix characterization (Hu et al., 2025) gives the deficit

$$D_{\mathrm{iid}} = \max_{0 \leq j \leq m} f_j, \qquad f_j = Q_j^n - P_j. \tag{4}$$

Theorem 1 (Optimal-IID exact attainment)**.** *For rational p, q and every $n \geq 1$, Optimal IID outputs $Y \sim p$ and attains*

$$\mathbb{P}(Y \in C(X)) = \alpha^*(p, q^{\otimes n}) = 1 - D_{\mathrm{iid}}. \tag{5}$$

Its preprocessing uses $O(V[\log(V+1) + \log(n+1)])$ arithmetic operations and $O(V)$ auxiliary storage; a prepared plan uses $O(n \log(V+1))$ operations per output. All score probabilities are rational, with polynomial finite-bit implementation under the input encoding in Appendix C.3.

The theorem turns the ceiling into an executable coupling, represented by cumulative arrays and a linear number of probability transfers.

An explicit selector marginal. Define the two cumulative envelopes and their meeting profile:

$$L_j = \max_{\ell \leq j} f_\ell, \quad U_j = \max_{\ell \geq j} f_\ell, \quad h_j = \min(L_j, U_j), \quad R_j = P_j + h_j, \quad \rho_j = R_j - R_{j-1}. \tag{6}$$

Set $\rho_i = 0$ when $q_i = 0$. The profile h rises to D_{iid} and then falls to the target mass outside $\mathrm{supp}(q)$; its total upward variation gives $\mathrm{TV}(p, \rho) = D_{\mathrm{iid}}$. The ratios ρ_i/q_i are nondecreasing, and the points (Q_j, R_j) lie above t^n. Interpolation then proves all coverage-base constraints, making ρ a feasible, TV-optimal selector marginal (Appendix A.2).

A sparse finite score realization. The largest rank in an IID pool has law $v_i = Q_i^n - Q_{i-1}^n$. We transform this rank rule into the desired selector through a symmetric kernel T satisfying

$$\sum_j T_{ij} = q_i, \qquad \sum_i T_{ij} = q_j, \qquad \sum_j T_{ij} \frac{v_j}{q_j} = \rho_i. \tag{7}$$

Pair the positive and negative masses of $\rho - v$ in cumulative order. Each match transfers probability between two ranks, giving at most $m - 1$ edges. Ordered ratios permit each row to be sampled by an edge choice and a rational switching coin (Appendix A.3, Lemma 9).

For each candidate $X_r = i$, independently draw J_r from $T_{i\cdot}/q_i$. The scores are IID with marginal q, so their maximum has law v. Conditional on the winning score j, the original token has posterior T_{ij}/q_j. The moment identity therefore gives its marginal ρ, and exact correction attains $1 - D_{\mathrm{iid}}$. This finite realization specializes moment-transfer ideas (Jourdain & Margheriti, 2020).

Algorithm 1 Optimal IID: exact sampling from an IID candidate pool

Require: Draft law q, target p, actual candidates $X_1, \ldots, X_n \sim q^{\otimes n}$
1: If $n = 1$, apply single-candidate maximal coupling and return
2: Sort $\mathrm{supp}(q)$ by p_i/q_i; form ρ using Equation (6)
3: Form $v_i = Q_i^n - Q_{i-1}^n$; pair the signed masses of $\rho - v$ into transfer edges
4: **for** each candidate position r **do**
5: Draw its finite score J_r using the edge and switching coins
6: **end for**
7: Choose a maximal-score position r^*, breaking ties independently of token identity
8: $Z \leftarrow X_{r^*}$; retain Z with probability $\min(1, p_Z/\rho_Z)$
9: On rejection, return a draw from $(p - \rho)_+/D_{\mathrm{iid}}$

The bottleneck moves. Theorem 1 attains the fixed-law upper bound, leaving no room for further acceptance gains under $q^{\otimes n}$. The next section changes the dependence among candidates while preserving each slot's marginal q.

4 Beyond IID through Stratified Proposals

4.1 Equal-mass stratification improves every subset's coverage

Keeping p, q, n fixed, we now coordinate the candidates across the draft probability mass. The idea is visible with two equally likely tokens and $n = 2$: drawing once from each half includes both tokens. For $p = (1, 0)$, this gives membership probability one, whereas the IID ceiling is $3/4$. Equal-mass stratification extends this coverage improvement to every q.

Choose a token order before the current target distribution is available. Place token i on a contiguous interval $I_i \subset [0, 1)$ of length q_i, and partition the unit interval into n equal strata $J_r = [(r-1)/n, r/n)$. Stratum r induces the categorical law

$$a_{ri} = n|I_i \cap J_r|. \tag{8}$$

Draw independently from the n rows a_r, then apply an independent uniform permutation to the slots. Since $\frac{1}{n}\sum_r a_{ri} = q_i$, every permuted slot has marginal q. The source labels are retained for verification. An interval may cross a stratum boundary, so repeated candidates remain possible.

Theorem 2 (Stratified coverage dominance)**.** *For every q, every $n \geq 1$, and every token subset S,*

$$c_{\mathrm{strat}}(S) = 1 - \prod_{r=1}^{n} (1 - a_r(S)) \;\geq\; 1 - (1 - q(S))^n = c_{\mathrm{iid}}(S). \tag{9}$$

Consequently, for every target p, $\alpha^*(p, \mathcal{Q}_{\mathrm{strat}}) \geq \alpha^*(p, q^{\otimes n})$.

The comparison holds against the *optimal* IID verifier for the same p, q, n. Each slot retains q, so every token's expected number of proposals remains nq_i. The gain comes from changing how those proposals jointly cover subsets.

Proof. Independence across strata gives the product in Equation (9). The factors have mean $1 - q(S)$ because $\sum_r a_r(S) = nq(S)$. The arithmetic–geometric mean inequality bounds their product by $(1 - q(S))^n$. Apply Equation (2). □

Coverage at the bottleneck. For a fixed average subset mass, the probability that all draws miss the subset is largest when the stratum masses are equal; unequal masses give strictly higher coverage. Acceptance improves strictly when coverage increases on every subset attaining the IID bottleneck deficit. Appendix B.1 derives this criterion from Equation (2).

4.2 An exact verifier for the improved proposal law

Recursive compression turns the coverage guarantee into an exact verifier. First construct a TV-optimal root marginal ρ. Then replace each half of the source pool by one actual candidate, choosing its marginal so that the parent's required selector remains feasible. Repeat on a balanced tree and correct the final representative to p. Interval geometry makes both the required marginals and these local selections tractable.

Interval geometry limits interactions. The token and stratum partitions have at most $V + n - 1$ nonempty intersections. A contiguous group of strata shares at most two boundary tokens with its exterior, and two adjacent child groups share at most one token. A balanced tree over strata can therefore represent each group using at most four boundary-inclusion states.

Principal curves summarize all relevant cuts. For a source group G and nonnegative token weights w, define

$$H_G(\lambda) = \max_S \{\lambda w(S) + 1 - c_G(S)\}, \qquad \lambda \geq 0. \tag{10}$$

The maximizing sets can be chosen as a nested chain, so each positive-weight token enters once and the curve has linear size. Independent groups multiply their failure probabilities; merging their exposed curves preserves the optimum for every parameter value. The resulting *balanced base* distributes each chain block's coverage increment proportionally to its token weights. With root weights $w = p$, this gives the required TV-optimal ρ.

Representatives preserve feasibility of the parent marginal. At a tree node, split any shared token's target mass between its two source copies. Replace each child pool by one representative drawn from its balanced base. All subset constraints needed to realize the current parent target remain feasible after this replacement. A finite two-source selector combines the representatives, and recursion realizes their prescribed marginals from the actual leaf candidates. This target-specific compression is what avoids enumerating candidate tuples. Appendix B gives the complete construction and proof.

Theorem 3 (Exact Stratified verification)**.** *For rational p, q and every $n \geq 1$, the recursive verifier outputs $Y \sim p$ and attains*

$$\mathbb{P}(Y \in C(X)) = \alpha^*(p, \mathcal{Q}_{\mathrm{strat}}). \tag{11}$$

Let $M = V + n$, $L = \lceil \log_2(M + 1) \rceil$, and $h = \lceil \log_2(n + 1) \rceil$. In the concurrent-read, exclusive-write (CREW) model, exact-arithmetic preprocessing uses $O(MLh^2)$ work, $O(L^2h^2)$ span, and $O(Mh)$ storage. A prepared plan uses $O(nL)$ work and $O(hL)$ span with exact probability coins. The same construction admits polynomial bit complexity.

Theorems 1–3 establish the constructive comparison

$$\boxed{\alpha_{\text{Optimal IID}} = \alpha^*(p, q^{\otimes n}) \;\leq\; \alpha_{\text{Stratified}} = \alpha^*(p, \mathcal{Q}_{\mathrm{strat}}).} \tag{12}$$

The plan is deterministic given p, q, n and the proposal order. During execution, fresh independent randomness selects representatives and performs the final correction. Appendix C analyzes the two nested tree traversals and the rational implementation.

Algorithm 2 Stratified proposals and exact recursive verification

Proposal stage, before observing p
Construct a_r from Equation (8); independently draw $X_r \sim a_r$
Randomly assign the draws to slots, retaining their source labels
Plan construction from p, q, n **and the proposal order**
Compute the root principal chain and its balanced base $\rho = \mathrm{Bal}_{c_{\mathrm{strat}}}(p)$
Recursively prepare the balanced source tree:
    at a leaf, store a rule returning that source's candidate
    at an internal node, split a shared token's mass, if present
        form child bases, store their two-source selector, and prepare both children
Online execution on candidates with retained source labels
Return leaf candidates; recursively apply stored selectors to child representatives
Obtain root token $Z \sim \rho$; apply maximal-coupling correction from ρ to p

5 The Remaining Dependence Frontier

To quantify how much room remains after optimal IID, allow arbitrary dependence between Y and the proposal slots while preserving $Y \sim p$ and $X_r \sim q$ for each r. This p-aware reference defines the Marginal Oracle.

Proposition 4 (Marginal-coupling optimum)**.** *The maximum membership probability over all such couplings is*

$$\alpha_{\mathrm{or}}(p, q, n) = \sum_i \min(p_i, nq_i). \tag{13}$$

The probability of successful output at token i is bounded by both p_i and its total proposal mass nq_i. Attainment assigns one mandatory occurrence to each successful token, fills the remaining slots from the unassigned proposal mass, and uniformly permutes the slots (Appendix D.1). The reference thus permits p to guide the proposal coupling, while Stratified forms its candidates from q before observing p.

For a collection of matched model states, we measure the fraction of remaining headroom captured by Stratified as

$$\mathcal{H} = \frac{\mathbb{E}[\alpha^*_{\mathrm{strat}} - \alpha^*_{\mathrm{iid}}]}{\mathbb{E}[\alpha_{\mathrm{or}} - \alpha^*_{\mathrm{iid}}]}, \tag{14}$$

with the same aggregation in numerator and denominator. This measures how much of the headroom to the target-aware endpoint is recovered by the prescribed Stratified proposal law.

Coverage also orders continuation values. For nonnegative token values v, define $\Phi_{\mathcal{Q}}(p, v) = \max_{X \sim \mathcal{Q}, Y \sim p} \mathbb{E}[v_Y \mathbf{1}\{Y \in C(X)\}]$. The feasible successful marginals are bounded by $x_i \le p_i$ and $x(S) \le c_{\mathcal{Q}}(S)$. Enlarging coverage enlarges this polytope. Therefore $\Phi_{\mathrm{strat}}(p, v) \ge \Phi_{\mathrm{iid}}(p, v)$ for every $v \ge 0$. Bellman induction extends this order to finite-horizon optimal continuation values under conditional local proposal laws and nonanticipating decisions. Here verification is optimized for the continuation weights. Appendices D.2–D.3 give the weighted characterization and continuation proof.

6 Experiments

Using matched language-model states, we evaluate the acceptance gained from dependence *after* IID verification is optimal. Numerical checks and scaling profiles assess the implementations; decoding latency provides supporting systems evidence.

Matched-state evaluation. We use Qwen3-4B/0.6B, Qwen3-32B/1.7B (Yang et al., 2025), and Gemma-2-27B/2B (Gemma Team et al., 2024) as target/draft pairs. The main Qwen3-4B study uses 344 prompts: 164 HumanEval, 100 GSM8K, and 80 MT-Bench prompts (Chen et al., 2021; Cobbe et al., 2021; Zheng et al., 2023). We collect 16 positions per prompt from target-only trajectories, yielding 5,504 matched states. The larger Qwen3 pair yields 1,376 states from four positions per

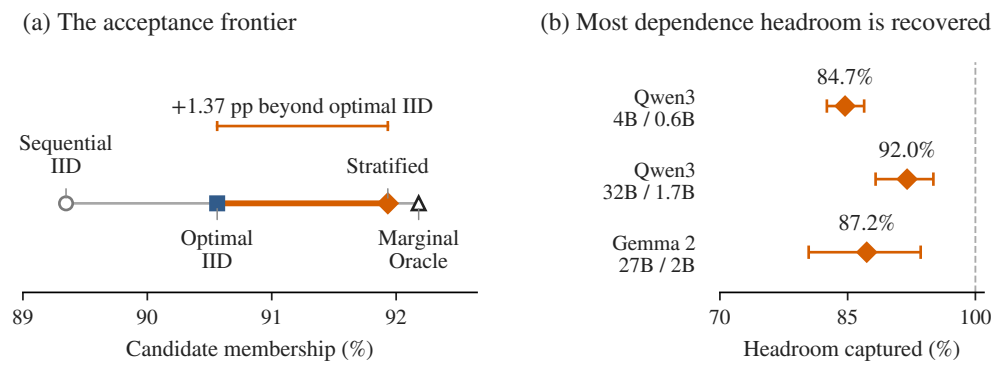

Figure 2: **Stratified recovers most of the acceptance headroom beyond optimal IID.** (a) The Qwen3-4B/0.6B frontier separates the verifier gain from the dependence gain. (b) Captured headroom for each model pair, with paired 95% prompt-cluster bootstrap intervals. All panels use $n = 4$, $K = 100$, temperature 1, and equal dataset weights. The Marginal Oracle is the p-aware reference in Equation (13); capture is the ratio of aggregate gaps in Equation (14).

Table 1: **Stratified surpasses the optimal IID ceiling on all three model pairs.** Candidate-membership probability (%), averaged equally across datasets, at $n = 4$, $K = 100$, temperature 1. The final column gives the paired gain over Optimal IID in percentage points, with a 95% interval. Gains are computed before rounding.

Target / draft	Sequential IID	Optimal IID	Stratified	Marginal Oracle	Gain over IID (pp)
Qwen3-4B/0.6B	89.35	90.56	91.93	92.18	1.37 [1.25, 1.50]
Qwen3-32B/1.7B	84.32	86.03	86.81	86.88	0.78 [0.65, 0.93]
Gemma-2-27B/2B	93.62	94.62	95.14	95.22	0.53 [0.37, 0.70]

prompt on the same prompt set. Gemma yields 600 states from 150 prompts, balanced across the three datasets. All methods at a state receive the same p, q; $n = 4$ and temperature 1 are the default. The top-K cutoff retains the K largest draft probabilities and renormalizes q; p remains over the full vocabulary.

Comparisons and uncertainty. Sequential IID uses sequential residual correction. Optimal IID and Stratified attain their respective fixed-law optima; the Marginal Oracle is Equation (13). We evaluate acceptance from the probability laws, average states within datasets, and weight datasets equally. Paired 95% intervals use 10,000 bootstrap samples of prompt clusters, keeping each prompt's positions together. Each model pair is evaluated separately (Appendix E).

Dependence gains across model pairs. Stratified exceeds optimal IID on all three pairs (Table 1). On Qwen3-4B/0.6B, Optimal IID raises membership from 89.35% to 90.56%; changing dependence raises it further to 91.93%. The latter gain recovers 84.7% of the remaining gap to the 92.18% oracle. The corresponding fractions are 92.0% for Qwen3-32B/1.7B and 87.2% for Gemma 2 (Figure 2). Full-vocabulary results give gains of 1.37 and 0.52 percentage points on the smaller Qwen3 pair and Gemma, respectively (Appendix E.4).

Candidate count and support. At $K = 100$, increasing n from one to ten raises both optimal ceilings (Figure 3). Stratified's absolute gain is 1.69 pp at $n = 2$ and 0.79 pp at $n = 10$; the captured headroom is 77.4% and 86.5%, respectively. At $n = 4$, varying draft support from $K = 50$ to the full vocabulary keeps the gain near 1.37 pp and the captured fraction between 84.5% and 84.9%. A separate $K = 100$ temperature study covers 384 states at each of $T \in \{0.6, 0.8, 1.0\}$; its gains remain positive throughout (Appendix E.5).

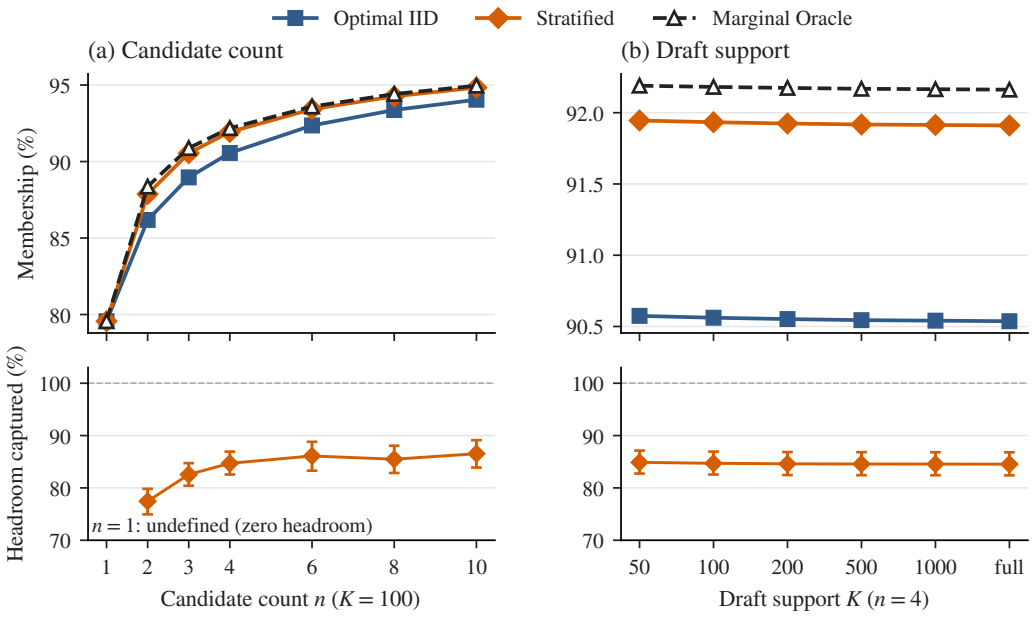

Figure 3: **The dependence gain persists across candidate counts and draft supports.** Qwen3-4B/0.6B on the same 5,504 states. Top: macro membership for the two optimal verifiers and the p-aware oracle. Bottom: captured headroom with prompt-cluster 95% intervals. At $n = 1$, all three values coincide and the capture fraction is undefined. Support settings are categorical; full denotes the 151,936-token vocabulary. Lines connect evaluated settings as visual guides.

Table 2: **Stratified improves throughput over AR and Sequential IID on one GH200.** Qwen3-32B/1.7B, $n = 4$, $K = 100$, depth one. Latencies are equal-dataset means. Speedups compare Stratified to the method listed in each row; brackets give paired prompt-cluster 95% intervals.

Method	Latency (ms/output token)	Stratified speedup [95% CI]
AR	21.241	1.5828 [1.5758, 1.5892]
Sequential IID	13.544	1.0092 [1.0063, 1.0122]
Stratified	13.421	1.0000 (reference)

Executing the optimal constructions. Explicit evaluation of finite couplings checks the samplers' target marginals and optimal membership probabilities (Appendix E). At $n = 4$, construction profiles reach 151,936 probability coordinates on synthetic inputs (Figure 5). Preparing a new proposal and a complete target-dependent plan takes 52.45 ms for Optimal IID on CPU and 34.89 ms for Stratified on A100 at that support. These device-specific profiles include rebuilding for new p, q and report prepared-plan output costs separately.

End-to-end feasibility on one GH200. We compare autoregressive sampling (AR), Sequential IID, and Stratified on Qwen3-32B/1.7B using one GH200, BF16 execution, batch size one, $n = 4$ depth-one candidates, $K = 100$, and temperature 1. Each method generates 128 tokens for all 344 prompts with two seeds. We time the complete decode loop, including proposal, verification, full-vocabulary residual correction, and KV updates. Stratified achieves 1.58× throughput relative to AR and 1.0092× relative to Sequential IID (Table 2). The pooled number of output tokens per round rises from 1.876 to 1.908, offsetting the increase in cost per round (Appendix E.7). A separate matched-state evaluation confirms that the production token order retains the gain beyond optimal IID (Appendix E.9).

Average verification cost below one millisecond per output token. A separate profiling run measures 0.489 ms/output-token for Stratified's verification-plan construction and online sampling, including residual correction, or 0.925 ms/verification round on average. Including proposal and target-support preparation gives 1.222 ms/output-token. The complete decoding times in Table 2 are measured without instrumentation; Appendix E.8 gives the disjoint cost breakdown.

7 Related Work

Exact sampling and multi-draft coupling. Speculative sampling uses rejection and residual correction to preserve the target distribution (Leviathan et al., 2023; Chen et al., 2023). The multi-draft transport formulation (Sun et al., 2023) and canonical selector-and-correction decomposition (Khisti et al., 2025) provide the foundations for our constructions.

From the IID value to an executable coupling. Hu et al.'s subset characterization gives the IID prefix formula used here (Hu et al., 2025). Global Resolution, our closest constructive predecessor, supplies overlapping selector quotas through its outer residual construction and approximates the coupling by convex optimization (Thomas & Pal, 2026). Appendix E.8 discusses its reported solve costs, success rates, and fallback. Optimal IID realizes an optimal selector through explicit finite scores, yielding exact attainment without iterative optimization.

Proposal dependence at fixed marginals. SpecHub studies joint sampling and verification to improve acceptance (Sun et al., 2024). Hu et al.'s greedy law fixes the top $n-1$ tokens and samples the last from the remainder. Having only one random candidate makes its optimal verifier tractable, but the slots do not retain marginal q (Hu et al., 2025, Section 5). Even this law's optimal acceptance can fall below the IID ceiling: Global Resolution reports higher acceptance for greedy at $K = 10$ and for optimal IID at the tested values $K \geq 100$ on Gemma-2 and Llama-3 (Thomas & Pal, 2026, Appendix O). List-level shared-randomness couplings provide acceptance lower bounds with conditional drafter-invariance guarantees (Rowan et al., 2025). Stratified combines fixed slot marginals, all-subset coverage dominance over IID, and exact attainment of its own fixed-law optimum.

8 Limitations

The exact samplers optimize membership for prescribed single-position proposal laws. Optimizing the proposal law or its token order remains a separate problem. Continuation-value dominance assumes value-aware verification; achieving the same complexity as the one-step sampler requires a further algorithmic construction.

Our exactness and complexity theorems use rational arithmetic; experiments use BF16 model execution and FP64 probability calculations validated on finite test cases.

Near-linear arithmetic complexity supports scalable exact verification, but does not by itself guarantee end-to-end acceleration. Practical gains depend on hardware, model choice, decoding configuration, and implementation efficiency, including GPU-kernel optimization. Our evaluation on one GH200, with batch size one and depth-one proposals, demonstrates feasibility and quantifies speedups in that setting.

9 Conclusion

Optimal IID turns the IID acceptance ceiling into an exact executable primitive. Stratified preserves the proposal marginals while changing dependence to improve subset coverage, and exactly attains the resulting fixed-law optimum. On the evaluated model states, it recovers approximately 85–92% of the headroom to the target-aware Marginal Oracle. Together, these results move the optimization focus from verification under IID proposals to tractable design of dependent proposals.

Reproducibility Statement

Appendices A–D specify the probability constructions, proofs, and computation models. Appendix E gives the evaluation protocols, aggregation, numerical checks, and software settings. The accompanying paper sources include the numerical summaries and scripts that regenerate the tables and vector figures.

AI Use Statement

AI tools were used to assist manuscript writing and language polishing, including improving clarity, grammar, wording, and the organization of the exposition. They were also used for literature retrieval and discovery, including identifying potentially relevant prior work and locating sources for further review. The authors are responsible for the final manuscript, including its scientific claims, arguments, and references.

References

Richard P. Brent and Paul Zimmermann. *Modern Computer Arithmetic*. Cambridge University Press, November 2010. ISBN 9780521194693. doi: 10.1017/cbo9780511921698. URL http://dx.doi.org/10.1017/CBO9780511921698.

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.

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.

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.

Shaddin Dughmi, David Kempe, and Ruixin Qiang. Alea Iacta Est: Auctions, Persuasion, Interim Rules, and Dice. In Avrim Blum (ed.), *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 31:1–31:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-095-8. doi: 10.4230/LIPIcs.ITCS.2019.31. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2019.31.

Satoru Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. Elsevier, second edition, 2005. URL https://www.kurims.kyoto-u.ac.jp/~fujishig/book-NH.html.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison,

Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving Open Language Models at a Practical Size, 2024. URL https://arxiv.org/abs/2408.00118.

Zhengmian Hu, Tong Zheng, Vignesh Viswanathan, Ziyi Chen, Ryan Rossi, Yihan Wu, Dinesh Manocha, and Heng Huang. Towards Optimal Multi-draft Speculative Decoding. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 3181–3203, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0907335ecf28faf15be54485dbcbe70e-Paper-Conference.pdf.

B. Jourdain and W. Margheriti. A new family of one dimensional martingale couplings. *Electronic Journal of Probability*, 25, 2020. ISSN 1083-6489. doi: 10.1214/20-ejp543. URL http://dx.doi.org/10.1214/20-EJP543.

Ashish Khisti, MohammadReza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. Multi-Draft Speculative Sampling: Canonical Decomposition and Theoretical Limits. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 1593–1633, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/04cdf500730af7733a6b13cbbc230206-Paper-Conference.pdf.

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.

Joseph Rowan, Truong Buu Phan, and Ashish Khisti. List-Level Distribution Coupling with Applications to Speculative Decoding and Lossy Compression. 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. 5576–5619. Curran Associates, Inc., 2025. doi: 10.52202/085713-0197. URL

https://proceedings.neurips.cc/paper_files/paper/2025/file/088bafc6da5f934cf9acbb361a71bfe5-Paper-Conference.pdf.

Christian Siebert and Jesper Larsson Träff. Perfectly load-balanced, optimal, stable, parallel merge, 2013. URL https://arxiv.org/abs/1303.4312.

Ryan Sun, Tianyi Zhou, Xun Chen, and Lichao Sun. SpecHub: Provable Acceleration to Multi-Draft Speculative Decoding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20620–20641, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1148. URL https://aclanthology.org/2024.emnlp-main.1148/.

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.

Rahul Thomas and Arka Pal. Global Resolution: Optimal Multi-Draft Speculative Sampling via Convex Optimization. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 81274–81307, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/83995c2acef585ae0f0d647154cfdd85-Paper-Conference.pdf.

Yepeng Weng, Qiao Hu, and Takehisa Yairi. UniVer: A Unified Perspective for Multi-step and Multi-draft Speculative Decoding, 2026. URL https://arxiv.org/abs/2605.04543.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

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.

A Coupling Foundations and the Optimal-IID Construction

All sets in the proofs are subsets of the finite token alphabet E. A candidate pool is nonempty, and multiplicities do not change its membership set C. Probabilities in this appendix are exact; the numerical implementation is described separately in Appendix E.

Proof outline. The proof of Theorem 1 has three steps. First, identify a candidate-selector marginal ρ whose total variation distance from p equals the fixed-law deficit. Second, realize that marginal by selecting an *original* candidate, using finite auxiliary scores. Third, verify the target marginal and the actual membership probability after correction. Lemmas 5–6 establish the shared coupling facts; Lemmas 7–8 construct ρ; Lemma 9 supplies the finite realization. Arithmetic costs are counted at the end of this appendix, and the corresponding bit complexity is analyzed in Appendix C.3.

A.1 Fixed-law optimum and candidate selectors

Lemma 5 (Coverage characterization)**.** *For any candidate law $\mathcal{Q}$, let $c(S) = \mathbb{P}(C \cap S \neq \varnothing)$. Then*

$$\alpha^*(p, \mathcal{Q}) = 1 - \max_S \{p(S) - c(S)\}. \tag{15}$$

A probability vector ρ can be realized by selecting an element of the actual pool if and only if $\rho(S) \leq c(S)$ for every S.

Proof. **Step 1: compute the maximum membership subflow.** Use one node for every candidate tuple x with positive mass $\mathcal{Q}(x)$ and one node for every token. Connect the source to x with capacity $\mathcal{Q}(x)$, connect x to each token in $C(x)$ with infinite capacity, and connect token i to the sink with capacity p_i. A flow assigns mass only to events on which the output is a candidate.

Fix the set S of token nodes on the sink side of a finite cut. Every tuple hitting S must also lie on that side, since otherwise an infinite-capacity edge crosses the cut. Every other tuple can be placed on the source side without increasing its capacity. The minimum capacity with this choice of S is therefore

$$\sum_{x: C(x) \cap S \neq \varnothing} \mathcal{Q}(x) + \sum_{i \notin S} p_i = c(S) + 1 - p(S). \tag{16}$$

Minimizing over S and applying max-flow/min-cut gives a maximum subflow of mass $a = 1 - \max_S(p(S) - c(S))$.

Step 2: complete the subflow to a correct coupling. Write f_{xi} for its membership-edge masses and let

$$u_x = \mathcal{Q}(x) - \sum_i f_{xi}, \qquad v_i = p_i - \sum_x f_{xi}. \tag{17}$$

Both vectors are nonnegative and sum to $1 - a$. If $a < 1$, add $u_x v_i/(1-a)$ to every tuple–token pair. This fills every remaining row and column, giving marginals $\mathcal{Q}$ and p. Whenever $u_x v_i > 0$, we have $i \notin C(x)$: otherwise the path from the source through x, i to the sink could carry another $\min(u_x, v_i) > 0$ units of flow. Thus the completion adds zero membership mass. If $a = 1$, no completion is needed.

Step 3: prove optimality and the selector statement. The membership events of any coupling with these marginals define a feasible subflow, so their total mass cannot exceed a. The preceding completion attains a, proving Equation (15). Replace the sink capacities p_i by a probability vector ρ_i. A flow of value one exists precisely when $1 - \rho(S) + c(S) \geq 1$ for all S, or $\rho(S) \leq c(S)$. Such a full flow defines a conditional choice of a token in each actual pool. Conversely, any such choice gives a full flow. This proves both directions of the selector assertion. □

The function c is normalized, monotone, and submodular: for a fixed nonempty set C, the indicator of hitting C has these properties, and averaging preserves them. Its base polytope is consequently the selector region used in the main text. The flow characterization is the finite form of the membership-transport formulation and its dual (Sun et al., 2023; Hu et al., 2025).

Lemma 6 (Exact correction of an optimal selector)**.** *Suppose $Z \in C$ has law $\rho \in \mathcal{B}(c)$ and $\mathrm{TV}(p, \rho) = D := \max_S(p(S) - c(S))$. Keep $Z = i$ with probability $\min(1, p_i/\rho_i)$; on rejection draw from $(p - \rho)_+/D$. The result has law p, membership probability $1 - D$, and zero membership mass on the rejection branch.*

Proof. Because p and ρ are probability vectors,

$$\sum_i (p_i - \rho_i)_+ = \sum_i (\rho_i - p_i)_+ = \mathrm{TV}(p, \rho) = D. \tag{18}$$

The mass retained at i is $\rho_i \min(1, p_i/\rho_i) = \min(p_i, \rho_i)$, with zero contribution when $\rho_i = 0$. Hence the rejection probability is $1 - \sum_i \min(p_i, \rho_i) = D$. For $D > 0$, an independent residual draw gives

$$\mathbb{P}(Y = i) = \min(p_i, \rho_i) + D\frac{(p_i - \rho_i)_+}{D} = p_i. \tag{19}$$

On retention, $Y = Z \in C$, so membership is at least $1 - D$. Lemma 5 gives the opposite bound for this correct coupling. Thus membership equals $1 - D$, and

$$\mathbb{P}(\text{reject},\ Y \in C) = \mathbb{P}(Y \in C) - \mathbb{P}(\text{retain}) = 0. \tag{20}$$

If $D = 0$, $p = \rho$ and retention is certain; the residual is not evaluated. This also explains why a zero-ρ coordinate never causes a division during execution. □

A.2 Prefix deficit and the two-sided envelope

Write $m = |\operatorname{supp}(q)|$ and relabel these tokens so that $a_i = p_i/q_i$ is nondecreasing. Let $p_0 = p(E \setminus \operatorname{supp}(q))$. For reference, all arrays used below are

$$Q_j = \sum_{i=1}^{j} q_i, \qquad P_j = \sum_{i=1}^{j} p_i, \qquad f_j = Q_j^n - P_j, \tag{21}$$

$$L_j = \max_{0 \le \ell \le j} f_\ell, \qquad U_j = \max_{j \le \ell \le m} f_\ell, \qquad h_j = \min(L_j, U_j), \qquad R_j = P_j + h_j. \tag{22}$$

In particular, $Q_0 = P_0 = f_0 = 0$, $Q_m = 1$, and $P_m = 1 - p_0$. The proposed selector has coordinates $\rho_i = R_i - R_{i-1}$ on this support and zero elsewhere.

Lemma 7 (Prefix reduction)**.** $\max_{T \subseteq E}(q(T)^n - p(T)) = \max_{0 \le j \le m}(Q_j^n - P_j)$.

Proof. Removing a zero-q token leaves $q(T)$ unchanged and can only decrease $p(T)$, so it suffices to optimize over the positive-q support.

For $t \in [0, 1]$, relax subset indicators to $z_i \in [0, 1]$ and define

$$F(t) = \min\left\{ \sum_i p_i z_i :\ \sum_i q_i z_i = t,\ 0 \le z_i \le 1 \right\}. \tag{23}$$

If $i < j$, $z_i < 1$, and $z_j > 0$, moving draft mass ε from j to i changes the objective by $\varepsilon(a_i - a_j) \le 0$. Repeating this exchange yields a prefix and at most one fractional coordinate. Therefore, for $t \in [Q_{j-1}, Q_j]$,

$$F(t) = P_{j-1} + a_j(t - Q_{j-1}). \tag{24}$$

Every subset is feasible in this relaxation at $t = q(T)$, so $q(T)^n - p(T) \le q(T)^n - F(q(T))$. On each displayed interval, $t^n - F(t)$ is convex for every integer $n \ge 1$. Its maximum is attained at an endpoint, even when $n = 1$ or ratios tie. Consequently

$$q(T)^n - p(T) \le \max_{0 \le j \le m}(Q_j^n - P_j). \tag{25}$$

Each endpoint corresponds to the actual subset $\{1, \dots, j\}$, which proves equality. □

Lemma 8 (Envelope feasibility and distance)**.** *The vector $\rho_i = R_i - R_{i-1}$ on $\operatorname{supp}(q)$, extended by zero elsewhere, is a probability vector. Its ratios ρ_i/q_i are nondecreasing, it belongs to $\mathcal{B}(c_{\mathrm{iid}})$, and $\mathrm{TV}(p, \rho) = D_{\mathrm{iid}}$.*

Proof. **Step 1: locate the envelope and its endpoints.** Let $g_j = Q_j^n$ and choose any index t attaining $D = D_{\mathrm{iid}} = \max_j f_j$. For $j \le t$, the suffix maximum is $U_j = D$, so $h_j = L_j$. For $j \ge t$, the prefix maximum is $L_j = D$, so $h_j = U_j$. Thus h is nondecreasing up to t and nonincreasing afterward, including any plateau of maximizers. Its endpoints are

$$h_0 = 0, \qquad h_t = D, \qquad h_m = f_m = p_0. \tag{26}$$

Both maxima defining h_j are at least f_j, giving $R_j \ge P_j + f_j = g_j$. Also $R_0 = 0$ and $R_m = 1$.

Step 2: prove the ratio ordering and nonnegativity. Put $x_i = \rho_i/q_i$ and $s_i = (g_i - g_{i-1})/q_i$. These s_i are successive secant slopes of t^n, hence $s_i \le s_{i+1}$. We check every adjacent pair $i, i+1$.

If $h_i > h_{i-1}$, the increase occurs on the prefix-max branch, where $L_i > L_{i-1}$ forces $L_i = f_i$. Thus $h_i = f_i$. If $h_{i+1} < h_i$, the decrease occurs on the suffix-max branch, where $U_i > U_{i+1}$ likewise forces $U_i = f_i$. In either case $R_i = g_i$. Using $R_{i-1} \ge g_{i-1}$ and $R_{i+1} \ge g_{i+1}$ then gives

$$x_i \le s_i \le s_{i+1} \le x_{i+1}. \tag{27}$$

In the remaining case, $h_i - h_{i-1} \le 0$ and $h_{i+1} - h_i \ge 0$. The identity $x_i = a_i + (h_i - h_{i-1})/q_i$ gives

$$x_i \le a_i \le a_{i+1} \le x_{i+1}. \tag{28}$$

These cases exhaust all pairs, including equality. Since $x_1 = R_1/q_1 \ge q_1^{n-1} > 0$, all coordinates of ρ on the support are positive. Their telescoping sum is $R_m - R_0 = 1$, so ρ is a probability vector.

Step 3: verify every subset constraint. Apply the fractional exchange argument from Lemma 7 with ρ in place of p. The ordering from Step 2 shows that the minimum ρ-mass at draft mass s is the interpolation $\tilde{R}(s)$ of (Q_j, R_j). On one interval write $s = (1-\theta)Q_{j-1} + \theta Q_j$. Then

$$\tilde{R}(s) = (1-\theta)R_{j-1} + \theta R_j \tag{29}$$

$$\ge (1-\theta)Q_{j-1}^n + \theta Q_j^n \ge s^n, \qquad 0 \le \theta \le 1. \tag{30}$$

The first inequality uses Step 1; the second is convexity. A subset therefore satisfies $\rho(S) \ge \tilde{R}(q(S)) \ge q(S)^n$. Zero-$q$ tokens have zero ρ and do not affect this argument. Applying it to the complement and using $\rho(E) = q(E) = 1$ gives

$$\rho(S) = 1 - \rho(E \setminus S) \le 1 - (1 - q(S))^n = c_{\mathrm{iid}}(S). \tag{31}$$

Thus ρ satisfies all base inequalities, including nonprefix subsets.

Step 4: compute the distance to the target. On positive-q coordinates, $\rho_i - p_i = h_i - h_{i-1}$. Step 1 implies

$$\sum_{i=1}^{m} (h_i - h_{i-1})_+ = \sum_{i=1}^{t} (h_i - h_{i-1}) = h_t - h_0 = D. \tag{32}$$

Outside the support, $\rho_i - p_i = -p_i \le 0$, so there is no additional positive variation. Both vectors have total mass one; hence $\mathrm{TV}(p, \rho) = \sum_i (\rho_i - p_i)_+ = D$. □

A.3 An ordered finite moment-transfer lemma

The following elementary form also supplies the two-source selector in Appendix B.7. It specializes the quantile-discrepancy construction of Jourdain & Margheriti (2020). Finite-score selection is also related to dice-based implementations of feasible interim rules (Dughmi et al., 2019); here the ordered moment constraints give an explicit sparse realization.

Lemma 9 (Sparse moment transfer)**.** *Let $w_i > 0$ sum to one, and let x_i, y_i be nondecreasing. Suppose $\delta_i = w_i(x_i - y_i)$ satisfies $\sum_{i \le j} \delta_i \ge 0$ for every j and $\sum_i \delta_i = 0$. There is a nonnegative symmetric matrix T with row and column sums w and moments $\sum_j T_{ij} y_j = w_i x_i$. Its off-diagonal support consists of at most $m - 1$ undirected edges. A kernel row can be sampled using one edge choice and one rational coin.*

Proof. **Step 1: construct the ordered transfer edges.** List the positive discrepancies in increasing index order, and do the same for the absolute values of the negative discrepancies. Their total masses agree. Overlay the cumulative intervals of the two lists; each nonempty intersection is a match of some mass $e > 0$ between a positive index i and a negative index j.

To check its direction, consider any point s in the interior of such an overlap. By index j, the negative cumulative mass has reached s. Prefix nonnegativity implies that the positive cumulative mass by that index is at least as large. The positive interval containing s must therefore have index at most j; the indices cannot be equal because their discrepancy signs differ. Hence $i < j$. Since x is nondecreasing, this gives

$$y_i < x_i \le x_j < y_j. \tag{33}$$

The score gap $y_j - y_i$ is consequently positive. Set $T_{ij} = T_{ji} = e/(y_j - y_i)$, sum repeated edges if any, and put the unused row mass on the diagonal. If there are m_+ positive and m_- negative discrepancies, the overlay has at most $m_+ + m_- - 1 \le m - 1$ intervals: each match exhausts at least one endpoint mass, and the last exhausts both. If every discrepancy is zero, simply take $T = \mathrm{diag}(w)$.

Step 2: verify nonnegative diagonals and both marginals. For a positive-discrepancy row i, every partner satisfies $y_j - y_i \ge x_i - y_i > 0$, so

$$\sum_{j \ne i} T_{ij} \le \frac{\sum_j e_{ij}}{x_i - y_i} = \frac{\delta_i}{x_i - y_i} = w_i. \tag{34}$$

For a negative row j, Equation (33) instead gives $y_j - y_i \ge y_j - x_j > 0$, and

$$\sum_{i \ne j} T_{ji} \le \frac{\sum_i e_{ij}}{y_j - x_j} = \frac{-\delta_j}{y_j - x_j} = w_j. \tag{35}$$

A zero-discrepancy row has no edges. The diagonal remainders are thus nonnegative and give row sums w. Symmetry then gives column sums w as well.

Step 3: verify the moment equations. Putting all row mass on its diagonal would give moment $w_i y_i$. For a matched edge from positive i to negative j, the off-diagonal entry changes these two moments by

$$T_{ij}(y_j - y_i) = e, \qquad T_{ji}(y_i - y_j) = -e. \tag{36}$$

The incident edge masses sum to the absolute discrepancy at each endpoint. Consequently

$$\sum_j T_{ij} y_j = w_i y_i + \delta_i = w_i x_i \tag{37}$$

for every row, including a row with zero discrepancy.

Step 4: sample a row without forming its diagonal. For $\delta_i \ne 0$, select an incident edge of mass e with probability $e/|\delta_i|$. Switch to its other endpoint j with probability

$$\frac{|x_i - y_i|}{|y_j - y_i|}; \tag{38}$$

otherwise stay at i. The gap bounds from Step 2 ensure that this probability lies in $[0, 1]$. Since $|\delta_i| = w_i |x_i - y_i|$, the probability of switching along this edge is

$$\frac{e}{|\delta_i|} \frac{|x_i - y_i|}{|y_j - y_i|} = \frac{e}{w_i |y_j - y_i|} = \frac{T_{ij}}{w_i}. \tag{39}$$

The remaining probability is T_{ii}/w_i. If $\delta_i = 0$, stay at i. This implementation stores only unnormalized edge masses and the switching quantities. Its representation is linear in the edge count, and it never sums separately divided off-diagonal entries to construct a diagonal. □

A.4 SCORES, ORIGINAL TOKENS, AND EXACT MEMBERSHIP

For $n \geq 2$, use Lemma 9 with $w = q$, $x_i = \rho_i/q_i$, $y_i = v_i/q_i$, and $v_i = Q_i^n - Q_{i-1}^n$. Strict convexity of t^n makes y_i strictly increasing. Lemma 8 gives nondecreasing x_i and

$$\sum_{i \leq j} (\rho_i - v_i) = R_j - Q_j^n \geq 0, \qquad \sum_i (\rho_i - v_i) = 0. \tag{40}$$

Thus all hypotheses hold. For each candidate, draw its score through the resulting row kernel using independent random bits. The joint distribution of original token and score is T, whose two marginals are both q.

The score law. The pairs (X_r, J_r) are independent over positions, each with joint mass T_{ij}. Summing over the original token gives $\mathbb{P}(J_r = j) = \sum_i T_{ij} = q_j$. Thus

$$\mathbb{P}\Big(\max_r J_r = j\Big) = \mathbb{P}(J_1, \ldots, J_n \leq j) - \mathbb{P}(J_1, \ldots, J_n < j) = Q_j^n - Q_{j-1}^n = v_j. \tag{41}$$

The original token at the winning position. For any score vector $\mathbf{j}$ of positive probability, independence of the pairs gives

$$\mathbb{P}(X_1 = i_1, \ldots, X_n = i_n \mid J = \mathbf{j}) = \prod_{r=1}^{n} \frac{T_{i_r j_r}}{q_{j_r}}. \tag{42}$$

Let W be a maximal-score position, chosen using only the score vector, the positions, and independent tie-breaking randomness. Conditional on $\mathbf{j}$ and $W = r$, the original token at that position still has posterior T_{ij_r}/q_{j_r}. Every eligible position has $j_r = \max_s j_s$, so averaging over W leaves that same posterior. This is why ties and repeated candidates do not alter the argument. Therefore the winning *original* token satisfies

$$\mathbb{P}(Z = i) = \sum_j v_j \frac{T_{ij}}{q_j} = \sum_j T_{ij} y_j = \rho_i. \tag{43}$$

It belongs to the actual pool. Lemmas 8 and 6 prove the distribution and membership assertions of Theorem 1.

Boundary cases. For $n = 1$, $\mathcal{B}(c)$ contains only q, and ordinary maximal coupling is used. Ratio ties may be resolved by any fixed order. Repeated candidates are retained as distinct positions during score selection. Zero-q target mass remains in the final residual. If $D = 0$, correction always retains; if $D = 1$, it always rejects and draws from p. No unreachable $0/0$ expression is evaluated.

Arithmetic cost. Ratio sorting costs $O(V \log(V + 1))$ operations. Computing all Q_j^n by repeated squaring costs $O(V \log(n + 1))$. Prefix/suffix maxima, cumulative discrepancy matching, and CDF construction have linear work and storage. Store the incident-edge CDF of each token, whose total size is linear, and the residual CDF. Each score uses binary search and a switching coin; one streaming maximum selects the winner. This gives $O(n \log(V + 1))$ online work and $O(V)$ auxiliary space, excluding the read-only candidate input. These costs prove the arithmetic part of Theorem 1.

B THE EXACT RECURSIVE STRATIFIED VERIFIER

Draws from the source distributions $a_1, \ldots, a_n$ in Equation (8) are independent before the slots are permuted. For a contiguous source group G, write

$$c_G(S) = 1 - \varphi_G(S), \qquad \varphi_G(S) = \prod_{r \in G} (1 - a_r(S)). \tag{44}$$

All constructions below use the distributions, weights, and fixed token order during preprocessing. Actual candidates enter only during execution of the resulting plan.

Proof outline. The recursive verifier needs a marginal that can be selected from each actual source group. We first compute the group's principal curve using interval geometry and a product merge. The curve gives a balanced base satisfying *all* subset inequalities. We then show that a feasible parent target remains feasible when its two independent child pools are replaced by categorical representatives. A scalar split handles their shared token, and an explicit two-source kernel selects one of the representatives. The final subsection combines these objects by induction. Appendix C analyzes this same construction.

B.1 From coverage improvement to acceptance gain

The cut formula identifies exactly which coverage improvements yield a strict acceptance gain. Let $d(S) = p(S) - c_{\mathrm{iid}}(S)$, $D = \max_S d(S)$, and $\Delta c(S) = c_{\mathrm{strat}}(S) - c_{\mathrm{iid}}(S) \geq 0$.

Lemma 10 (Bottleneck-gap identity)**.** *The optimal acceptance gain satisfies*

$$\alpha^*(p, \mathcal{Q}_{\mathrm{strat}}) - \alpha^*(p, q^{\otimes n}) = \min_{S \subseteq E} \{D - d(S) + \Delta c(S)\}. \tag{45}$$

It is strictly positive if and only if $\Delta c(S) > 0$ for every subset attaining $d(S) = D$.

Proof. By Lemma 5, the two deficits are D and $\max_S(d(S) - \Delta c(S))$. Subtracting the corresponding acceptances gives

$$\alpha^*_{\mathrm{strat}} - \alpha^*_{\mathrm{iid}} = D - \max_S \{d(S) - \Delta c(S)\} \tag{46}$$

$$= \min_S \{D - d(S) + \Delta c(S)\}. \tag{47}$$

For every S, both $D - d(S)$ and $\Delta c(S)$ are nonnegative. The displayed summand is zero exactly when $d(S) = D$ and $\Delta c(S) = 0$. There are finitely many subsets, so the minimum is positive if and only if no subset satisfies both equalities. This proves the criterion, including cases with several maximizing cuts. □

In the stratified construction, $\Delta c(S) > 0$ exactly when the stratum masses $a_r(S)$ are not all equal: this is the strictness condition of the arithmetic–geometric mean inequality in Theorem 2. A fixed token order determines these masses while preserving coverage dominance for every target. The criterion therefore explains how the order can affect the magnitude of the gain while ensuring that the gain remains nonnegative.

B.2 Geometry and conditional principal curves

Lemma 11 (Interval geometry)**.** *There are at most $V + n - 1$ nonempty token–stratum intersections. A contiguous source group shares at most two distinct tokens with its exterior; adjacent child groups share at most one token.*

Proof. Overlay the token partition and the stratum partition of $[0, 1)$. Their interior breakpoints form at most $(V - 1) + (n - 1)$ distinct points, giving the incidence bound. A token appearing on both sides of a source-group boundary must cross that boundary. There is at most one such token at either endpoint and one at the split between adjacent children. A long token crossing both endpoints is still a single token. □

Build a balanced binary tree over the ordered sources. Assign each token's weight to the lowest common ancestor of all its nonempty incidences, called its owner. The token contributes to each applicable failure factor but contributes its weight exactly once. At a group boundary, store inclusion bits for its at most two distinct exterior-shared tokens. Thus there are at most four boundary masks per group. If the two boundaries belong to one token, they use the same bit.

For a nonnegative weight vector w and a fixed legal boundary mask, maximize

$$\lambda w(S) + \varphi_G(S), \quad \lambda \geq 0, \tag{48}$$

over internal choices, retaining forced boundary tokens in the coverage calculation. The owner rule supplies the weight in the appropriate node. Free zero-weight tokens are omitted; any required

zero-weight boundary token stays fixed. Represent the optimal value by the upper exposed hull of points $(u, v) = (w(S), \varphi_G(S))$.

Lemma 12 (Nested exposed choices)**.** *For each fixed boundary mask, the free positive-weight maximizing sets in Equation* (48) *can be selected as a nested chain. Its number of exposed vertices is at most one plus the number of such tokens.*

Proof. **Step 1: compare minimizers at two parameters.** The equivalent minimization objective is $c_G(S) - \lambda w(S)$. Fixing a boundary mask leaves a family closed under union and intersection, and the restricted coverage is submodular. At a fixed parameter its minimizers are also closed under union: if two sets attain the same minimum, submodularity forces their union and intersection to attain it as well. Finiteness therefore gives an inclusion-wise maximal minimizer.

For $\lambda_1 < \lambda_2$, take these maximal minimizers S_1, S_2 and abbreviate $I = S_1 \cap S_2$, $U = S_1 \cup S_2$. Their optimality gives

$$c_G(S_1) - \lambda_1 w(S_1) \le c_G(I) - \lambda_1 w(I), \tag{49}$$

$$c_G(S_2) - \lambda_2 w(S_2) \le c_G(U) - \lambda_2 w(U). \tag{50}$$

Adding and using the modularity of w yields

$$0 \le c_G(S_1) + c_G(S_2) - c_G(I) - c_G(U) \tag{51}$$

$$\le -(\lambda_2 - \lambda_1) w(S_1 \setminus S_2). \tag{52}$$

The first inequality is submodularity. Hence

$$(\lambda_2 - \lambda_1) w(S_1 \setminus S_2) \le 0. \tag{53}$$

Step 2: extract and count the exposed choices. No free positive-weight token can leave as the parameter increases. A free zero-weight token can be removed without increasing $c_G - \lambda w$, because coverage is monotone; we consistently omit it. Forced zero-weight boundary tokens remain included as prescribed. Thus the free choices can be represented by a nested chain. At a breakpoint, retaining its two endpoint choices suffices to represent the value on both sides; it is unnecessary to store every tied subset. Each change adds at least one previously absent positive-weight token, so the number of exposed vertices is at most one plus the number of those tokens. □

At a leaf, failure has the form $b - \sum_{i \in S} a_{ri}$ after accounting for fixed tokens. A free token enters when $\lambda w_i \ge a_{ri}$. Sorting by decreasing w_i / a_{ri} therefore gives the leaf curve, with coincident breakpoints grouped. Coverage on a single source has a unique full base, namely its categorical distribution.

B.3 A LINEAR-SIZE PRODUCT-CURVE MERGE

Consider two child hulls with ordered vertices (u_j, v_j), increasing u and decreasing v. Remove duplicate points, keep the largest v at a fixed u, and omit dominated horizontal points. For each remaining segment set

$$d_j = \frac{u_j - u_{j-1}}{v_{j-1} - v_j}, \qquad \tau_j = d_j v_j. \tag{54}$$

Upper-hull concavity makes d_j nonincreasing, and hence also τ_j. Merge the two τ sequences in decreasing order, stably resolving ties. Starting from the initial pair, advance the indicated child at each event and record

$$(u_i^L + u_j^R,\ v_i^L v_j^R). \tag{55}$$

Take the upper exposed hull of this list.

Lemma 13 (Product merge)**.** *The list in Equation* (55) *contains at most one more point than the total number of child segments and preserves the optimum of $\lambda(u^L + u^R) + v^L v^R$ for every $\lambda \ge 0$.*

Proof. **Step 1: restrict attention to child hull vertices.** For a fixed right point, the objective is $\lambda u^L + v^R v^L$ plus a constant. Its coefficients are nonnegative, so a left exposed endpoint attains its maximum; the same argument applies on the right. A horizontal tie is resolved by retaining its largest u. Thus discarding nonexposed or dominated child points preserves an optimal pair, including the case where one coefficient is zero.

Step 2: identify the selected segments of an optimum. Fix $\lambda > 0$ and choose an optimal pair (i, j) with largest total u among ties. If $v_i^L v_j^R = 0$, advancing both children to their largest u preserves the zero product and can only increase the objective. The final point in the merged list therefore suffices. Otherwise put $v^* = v_i^L v_j^R > 0$.

With the right vertex fixed, advancing left segment a changes the objective by

$$\lambda(u_a^L - u_{a-1}^L) + v_j^R(v_a^L - v_{a-1}^L) \tag{56}$$

$$= (v_{a-1}^L - v_a^L)(\lambda d_a^L - v_j^R). \tag{57}$$

All failure drops in this expression are positive after preprocessing. Since d_a^L is nonincreasing, these increments change sign at most once. Optimality requires $\lambda d_i^L \geq v_j^R$ for the last selected segment, and the maximal-u convention requires $\lambda d_{i+1}^L < v_j^R$ for the first unselected segment, when those segments exist. Monotonicity extends these inequalities to all selected and unselected segments, respectively.

Step 3: recover the optimum from a single merged prefix. For a selected left segment $a \leq i$, we also have $v_a^L \geq v_i^L$. For an unselected segment $a > i$, $v_a^L \leq v_i^L$. Therefore

$$a \leq i \implies \tau_a^L = d_a^L v_a^L \geq \frac{v_j^R v_i^L}{\lambda} = \frac{v^*}{\lambda}, \tag{58}$$

$$a > i \implies \tau_a^L < \frac{v^*}{\lambda}. \tag{59}$$

The same argument applies to the right child. Hence the selected segments of both children are exactly the prefix of the global decreasing τ order above this common threshold. Every segment tied at the threshold is selected; stable resolution of other ties is harmless. The merge records the associated pair (i, j).

Step 4: cover the remaining parameter and count the list. At $\lambda = 0$, maximizing the failure product uses the initial maximum-failure points. The merged list thus contains an optimum for every $\lambda \geq 0$. Every recorded point is itself an admissible child pair, so taking the upper hull of the list cannot overestimate the optimum either. Starting with one point and advancing one child segment per event gives exactly one plus the total number of child segments before any deduplication. □

If the children share token h, enumerate its two inclusion states, require consistency of its two incidences, and add w_h once at its owner. For each legal parent mask there is only a constant number of child-mask combinations. Apply Lemma 13 to each, combine the resulting lists, and take a hull. The maximization over compatible masks is exactly the parent objective. Induction from the leaves proves the correctness of the complete principal query.

More explicitly, every legal parent subset restricts to one of these compatible child-mask pairs. Conversely, any pair of child selections with consistent shared-token bits forms a legal parent subset. Independence gives the product of child failure probabilities; the owner convention makes their accumulated weights add to the parent weight. Thus the merge has neither omitted a feasible selection nor introduced an incompatible one. The at-most-two exterior tokens and at-most-one interior shared token keep this enumeration constant at each node.

Recovering the chain. A hull vertex stores a selection reference: a prefix length at a leaf, or child reference IDs and the local shared-token bit at an internal node. Canonical IDs are assigned to constant-length internal keys by sorting and scanning. The exterior boundary mask is retained separately; it is part of the full conditional selection. To recover an adjacent chain difference, descend only into unequal references and include any changed local token. Identical subtrees are skipped. The expanded references can be charged to newly added tokens and their ancestors, costing $O(M_G \log(n_G + 1))$ over the chain, where M_G is the incidence count and n_G the source count. This gives a sparse representation of the entire chain without storing a full token vector at every vertex.

B.4 BALANCED BASES AND ALL-SUBSET FEASIBILITY

For a full group query with positive-weight support W_+, recover $\varnothing = S_0 \subset S_1 \subset \cdots \subset S_t = W_+$. Write $U_j = S_j \setminus S_{j-1}$ and set

$$\lambda_j = \frac{c_G(S_j) - c_G(S_{j-1})}{w(U_j)}, \qquad \beta_i = \lambda_j w_i \quad (i \in U_j). \tag{60}$$

The breakpoints λ_j are nonnegative and increasing after equal slopes are combined. Extend to zero-weight tokens $z_1, z_2, \ldots$ in the fixed token order by the contraction greedy rule

$$\beta_{z_j} = c_G(W_+ \cup \{z_1, \ldots, z_j\}) - c_G(W_+ \cup \{z_1, \ldots, z_{j-1}\}). \tag{61}$$

We denote this deterministic base by $\mathrm{Bal}_{c_G}(w)$; the definition includes the all-zero vector through Equation (61).

Lemma 14 (Balanced-base properties)**.** *The vector $\beta = \mathrm{Bal}_c(w)$ is a base of c. Let $\mathcal{L}_w$ be the piecewise-linear interpolation of $(w(S_j), c(S_j))$ on the positive-weight chain, with $\mathcal{L}_w(0) = 0$. Then, for every T and every $\lambda \geq 0$,*

$$\beta(T) \geq \mathcal{L}_w(w(T)), \tag{62}$$

$$\min_T \{c(T) - \lambda w(T)\} = \min_T \{\beta(T) - \lambda w(T)\}. \tag{63}$$

For $w = p$ restricted to the proposal support, $\mathrm{TV}(p, \beta) = \max_T (p(T) - c(T))$ after extending β by zero outside that support.

Proof. **Step 1: prove feasibility on an arbitrary positive-weight subset.** Fix $T \subseteq W_+$ and write $T_j = T \cap U_j$, $T_{<j} = T \cap S_{j-1}$. Both S_{j-1} and S_j minimize $c(S) - \lambda_j w(S)$ at their shared breakpoint. Comparing S_{j-1} to $S_{j-1} \cup T_j$ gives

$$c(S_{j-1}) - \lambda_j w(S_{j-1}) \leq c(S_{j-1} \cup T_j) - \lambda_j w(S_{j-1} \cup T_j), \tag{64}$$

and therefore

$$\lambda_j w(T_j) \leq c(S_{j-1} \cup T_j) - c(S_{j-1}). \tag{65}$$

Since $T_{<j} \subseteq S_{j-1}$ and T_j is disjoint from S_{j-1}, diminishing marginal returns gives

$$c(S_{j-1} \cup T_j) - c(S_{j-1}) \leq c(T_{<j} \cup T_j) - c(T_{<j}). \tag{66}$$

Summing the two bounds over j telescopes:

$$\beta(T) = \sum_j \lambda_j w(T_j) \leq \sum_j [c(T \cap S_j) - c(T \cap S_{j-1})] = c(T). \tag{67}$$

Step 2: include zero-weight tokens and the total mass. Let $Z_{<j} = \{z_1, \ldots, z_{j-1}\}$. If $z_j \in T$, its greedy increment is bounded by

$$\beta_{z_j} = c(W_+ \cup Z_{<j} \cup \{z_j\}) - c(W_+ \cup Z_{<j}) \tag{68}$$

$$\leq c((T \cap W_+) \cup (T \cap Z_{<j}) \cup \{z_j\}) - c((T \cap W_+) \cup (T \cap Z_{<j})). \tag{69}$$

Again this is diminishing marginal returns. Summing only over the zero-weight tokens of T proves $\beta(T \setminus W_+) \leq c(T) - c(T \cap W_+)$. Together with Step 1, this proves $\beta(T) \leq c(T)$ for every subset. All increments are nonnegative, and along the full chain their sum is

$$\beta(E) = c(W_+) + [c(E) - c(W_+)] = c(E) = 1. \tag{70}$$

The construction also gives $\beta(S_j) = c(S_j)$ at every chain endpoint. These assertions include $W_+ = \varnothing$, when only the greedy extension is used.

Step 3: prove the interpolation bound and parameter identity. The nondecreasing ratios $\beta_i/w_i = \lambda_j$ mean that minimizing β-mass subject to a fixed positive weight mass x fills the blocks in chain order, with at most one fractional block. Its value is exactly $\mathcal{L}_w(x)$. A subset has at least that mass, and any selected zero-weight tokens add a nonnegative amount. This proves Equation (62).

For the parameter identity, feasibility gives

$$\min_T [c(T) - \lambda w(T)] \geq \min_T [\beta(T) - \lambda w(T)]. \tag{71}$$

The latter objective is the sum of coordinate contributions. Include all blocks with $\lambda_j < \lambda$, exclude all blocks with $\lambda_j > \lambda$, and take an endpoint choice when $\lambda_j = \lambda$. Exclude the zero-weight coordinates. The resulting set S_* is a chain endpoint and satisfies $\beta(S_*) = c(S_*)$, so

$$\min_T [c(T) - \lambda w(T)] \leq c(S_*) - \lambda w(S_*) = \min_T [\beta(T) - \lambda w(T)]. \tag{72}$$

This proves Equation (63) in both directions.

Step 4: specialize to the target and compute total variation. Let E_q be the proposal support and $p_0 = p(E \setminus E_q)$. Adding an outside-support token to a cut changes its target mass but not coverage. A maximizing cut can therefore include all such tokens, giving

$$\max_{S \subseteq E} (p(S) - c(S)) = p_0 + \max_{T \subseteq E_q} (p(T) - c(T)) \tag{73}$$

$$= p_0 + \max_{T \subseteq E_q} (p(T) - \beta(T)) \tag{74}$$

$$= p_0 + \sum_{i \in E_q} (p_i - \beta_i)_+ = \mathrm{TV}(p, \beta). \tag{75}$$

The second line uses Step 3 at $\lambda = 1$ with $w = p|_{E_q}$; the third maximizes a modular set function coordinatewise. Since β is zero outside E_q and both full vectors have mass one, the last expression is precisely their total variation distance. □

This is a specialized principal-partition construction; the general submodular background is described by Fujishige (2005). The interval query makes its chain computable at the stated cost.

Lemma 15 (Recognizing a balanced base from a tight chain)**.** *Let $b \in \mathcal{B}(c)$ and $w \geq 0$. Suppose a chain from $\varnothing$ to $W_+ = \mathrm{supp}(w)$ is tight for b, and on each of its blocks $b_i = \gamma_j w_i$ with nonnegative, nondecreasing γ_j. Suppose also that the zero-weight coordinates of b are the fixed-order greedy extension of this chain. Then $b = \mathrm{Bal}_c(w)$.*

Proof. For each parameter λ, a minimizer of $b(T) - \lambda w(T)$ is the prefix containing precisely the blocks with $\gamma_j < \lambda$, with endpoint choices at equality. That prefix is tight, and $c(T) \geq b(T)$ for all T. The two inequalities in Step 3 of Lemma 14 therefore show that this prefix also minimizes $c(T) - \lambda w(T)$.

When λ differs from every γ_j, the minimizing positive-weight coordinates of the modular objective are unique. Any coverage minimizer must also minimize that modular objective, since their minimum values agree. Hence the positive-weight principal chain has exactly these blocks, merging adjacent equal slopes. Tightness gives

$$\gamma_j = \frac{b(S_j) - b(S_{j-1})}{w(S_j \setminus S_{j-1})} = \frac{c(S_j) - c(S_{j-1})}{w(S_j \setminus S_{j-1})}. \tag{76}$$

Thus Equation (60) reconstructs b on W_+. The stipulated zero-extension rule reconstructs the remaining coordinates. If $w = 0$, this last rule alone proves the assertion. □

B.5 Target-preserving compression of independent groups

Lemma 16 (Two-group representative compression)**.** *Suppose two nonempty independent pools have disjoint label sets and coverages c_L, c_R. Let nonnegative target parts w_L, w_R sum to one and satisfy, for every S, T,*

$$w_L(S) + w_R(T) \leq c_R(T) + (1 - c_R(T)) c_L(S). \tag{77}$$

If $\beta_L = \mathrm{Bal}_{c_L}(w_L)$ and $\beta_R = \mathrm{Bal}_{c_R}(w_R)$, the same target is feasible for two independent categorical representatives with laws β_L, β_R.

Proof. **Step 1: replace the left group while holding the right group fixed.** Fix an arbitrary right subset T and put $v = c_R(T)$. If $v < 1$, define the affine function

$$\phi_T(x) = \frac{x + w_R(T) - v}{1 - v} \tag{78}$$

Equation (77), evaluated at each left chain endpoint S_j, gives $\phi_T(w_L(S_j)) \le c_L(S_j)$. This also holds at the empty endpoint because $w_R(T) \le v$. On every segment between consecutive endpoints, the affine function stays below their linear interpolation. By Equation (62), for every left subset S,

$$\beta_L(S) \ge \mathcal{L}_{w_L}(w_L(S)) \ge \frac{w_L(S) + w_R(T) - v}{1 - v}. \tag{79}$$

Multiplying by the positive denominator yields

$$w_L(S) + w_R(T) \le v + (1 - v)\beta_L(S). \tag{80}$$

The right side is exactly the coverage of $S \cup T$ for a left categorical representative and the original right pool.

Step 2: check the cases in which that division is unavailable. If $v = 1$, the new coverage is one and Equation (80) follows from total target mass one. If $w_L = 0$ and $v < 1$, the only weight argument is $x = 0$: the empty-set constraint gives $\phi_T(0) \le 0$, while $\beta_L(S) \ge 0$. Thus the same conclusion holds with the prescribed greedy extension for $\mathrm{Bal}_{c_L}(0)$.

Step 3: replace the right group. Equation (80) now holds for every S, T, so the once-replaced system meets the hypothesis needed for the same argument with left and right exchanged. Holding the left categorical representative fixed gives

$$w_L(S) + w_R(T) \le \beta_L(S) + (1 - \beta_L(S))\beta_R(T) = 1 - (1 - \beta_L(S))(1 - \beta_R(T)). \tag{81}$$

This is every subset constraint for the two independent representatives. Their target masses sum to one, so Lemma 5 proves selector feasibility. The argument performs two successive valid replacements; it does not assume the conclusion of the second replacement when proving the first. □

B.6 Splitting the single shared token

Suppose a feasible parent target r contains a token h shared by its two children. Give its left and right incidences distinct labels h_L, h_R. With split mass t, assign target masses $t, r_h - t$ to these labels and retain the other coordinates. Let r^0 denote the reference $t = 0$. For the independent lifted pools, define the deficit $d(S) = r^0(S) - c_{LR}(S)$. Let C_+ and C_- be its maxima over cuts containing only h_L and only h_R, respectively. Cuts containing both or neither are compatible with original-token cuts and have deficit at most zero.

Lemma 17 (Feasible shared-token split)**.** *The value $t = \max(0, C_-)$ belongs to $[0, r_h]$ and makes the lifted target feasible for every cut.*

Proof. **Step 1: bound each one-copy deficit.** If S contains only h_L, let $S' = S \setminus \{h_L\}$. Since $r^0_{h_L} = 0$ and coverage is monotone,

$$d(S) = r^0(S') - c_{LR}(S) \le r^0(S') - c_{LR}(S') = d(S') \le 0. \tag{82}$$

The last cut contains neither copy and corresponds to an original-token cut. Maximizing gives $C_+ \le 0$. For an only-right cut, delete h_R instead; its target mass is r_h, so

$$d(S) \le d(S \setminus \{h_R\}) + r_h \le r_h. \tag{83}$$

Thus $C_- \le r_h$.

Step 2: relate the two extrema by uncrossing. Finiteness gives maximizing sets S_+, S_-. The function $d = r^0 - c_{LR}$ is supermodular, hence

$$C_+ + C_- = d(S_+) + d(S_-) \le d(S_+ \cup S_-) + d(S_+ \cap S_-) \le 0. \tag{84}$$

The union contains both copies and the intersection contains neither. Both correspond to original-token cuts, which explains the last inequality; feasibility of arbitrary one-copy cuts has not been assumed.

Step 3: check all four cut classes after the split. Let r^t be the split target. Changing from r^0 to r^t adds t to the left copy and subtracts t from the right copy. The four possibilities are

Copies included	$r^t(S) - c_{LR}(S)$	Required bound
Neither	$d(S)$	$d(S) \le 0$
Both	$d(S)$	$d(S) \le 0$
Only h_L	$d(S) + t$	$C_+ + t \le 0$
Only h_R	$d(S) - t$	$C_- - t \le 0$

If $C_- \le 0$, choose $t = 0$ and all four bounds follow from Step 1. If $C_- > 0$, choose $t = C_-$; then Step 1 gives $0 < t \le r_h$, and Step 2 gives $C_+ + t \le 0$. The only-right bound is equality at its maximum. Thus $t = \max(0, C_-)$ is a nonnegative split no larger than r_h, preserves total target mass, and satisfies every lifted coverage constraint. □

The conditional principal queries compute these extrema: with a prescribed inclusion state, maximize $r^0(S) + \varphi_{LR}(S) - 1$ by evaluating the merged curve at $\lambda = 1$. Only the two states of the local shared label are involved. After the split, Lemma 16 applies to the disjoint lifted label sets.

B.7 A FINITE SELECTOR FOR TWO CATEGORICAL SOURCES

Let the independent representatives have laws a_i and b_t on disjoint labels, restricted to their positive supports, and let the desired output masses be l_i, r_t, totaling one and satisfying all two-source coverage inequalities. Write $m = |\operatorname{supp}(a)|$ and $m_{\mathrm{LR}} = |\operatorname{supp}(a)| + |\operatorname{supp}(b)|$. Sort the left ratios $x_i = l_i/a_i$ increasingly. Singleton feasibility gives $0 \le x_i \le 1$. Set $x_0 = 0$, $x_{m+1} = 1$, and $A_j = \sum_{i \le j} a_i$ including $A_0 = 0$.

Assign an auxiliary right score $J \in \{0, \dots, m\}$ with marginal

$$\pi_j = x_{j+1} - x_j. \tag{85}$$

Select the left token i when $i > J$, and the right token otherwise. The left output mass is $a_i \mathbb{P}(J < i) = a_i x_i = l_i$. Given a right token t, its winning probability is $\mathbb{E}[A_J \mid t]$, so it remains to realize

$$\mathbb{E}[A_J \mid t] = f_t := r_t / b_t. \tag{86}$$

Lemma 18 (Finite right-score kernel)**.** *The feasibility constraints imply a rational kernel with column marginal π satisfying Equation* (86)*. It has a linear-size sampling representation, $O(m_{\mathrm{LR}} \log(m_{\mathrm{LR}} + 1))$ construction work, and $O(\log(m_{\mathrm{LR}} + 1))$ work per selection, where m_{LR} is the combined support size.*

Proof. **Step 1: derive the required upper-tail inequality.** For a right subset S of mass $u = b(S)$, joint feasibility for every left subset U gives

$$\sum_{i \in U} a_i x_i + r(S) \le a(U) + u - a(U)u. \tag{87}$$

Rearrange and minimize over U. Each left coordinate may be included independently in this minimization, so

$$r(S) \le u + \min_U \sum_{i \in U} a_i (1 - u - x_i) \tag{88}$$

$$= \sum_i a_i \min(u, 1 - x_i). \tag{89}$$

For the second equality, write the first expression as $\sum_i a_i [u + \min(0, 1 - u - x_i)]$. Now let $H_A(u)$ denote the upper-quantile integral over mass u of the law placing mass π_j at A_j. We have

$$A_J = \sum_i a_i \mathbf{1}\{J \ge i\}, \qquad \mathbb{P}(J \ge i) = \sum_{j=i}^{m} (x_{j+1} - x_j) = 1 - x_i. \tag{90}$$

These events are nested upper tails of J. In its largest mass-u portion, the event $\{J \ge i\}$ occupies mass $\min(u, 1 - x_i)$. Integrating the displayed sum therefore gives $H_A(u) = \sum_i a_i \min(u, 1 - x_i)$. Equation (89) is consequently the bound $r(S) \le H_A(b(S))$.

Step 2: extend the subset bound to all quantile masses. Sort right tokens by nondecreasing f_t, and let $H_f(u)$ be the upper-quantile integral of the law placing mass b_t at f_t. At an upper suffix S, its mass is $b(S)$ and its integral is $r(S)$. Step 1 therefore gives $H_f(u) \le H_A(u)$ at every suffix endpoint, including zero and one. Between consecutive endpoints H_f is linear, while H_A is concave, so this inequality holds throughout $[0, 1]$.

The means agree, since

$$\sum_t b_t f_t = \sum_t r_t = 1 - \sum_i l_i = \sum_i a_i(1 - x_i) = \sum_j \pi_j A_j. \tag{91}$$

Consequently their *lower*-quantile integrals satisfy $L_f(s) \ge L_A(s)$ for every s: subtract the upper-tail inequality at mass $1 - s$ from the common mean. This is the exact inequality needed by the moment-transfer construction.

Step 3: construct the finite kernel on common atoms. Overlay the cumulative b_t intervals, ordered by f_t, with the cumulative π_j intervals, ordered by A_j. Discard zero-length intervals. The at most $m_{\mathrm{LR}} + 1$ remaining atoms have lengths $w_s > 0$, desired means $\xi_s = f_{t(s)}$, and score values $\eta_s = A_{j(s)}$. Both sequences are nondecreasing, and Step 2 gives, at every atom endpoint z,

$$\sum_{s \le z} w_s(\xi_s - \eta_s) = L_f\left(\sum_{s \le z} w_s\right) - L_A\left(\sum_{s \le z} w_s\right) \ge 0. \tag{92}$$

The total discrepancy is zero. Lemma 9 therefore produces a matrix T with row and column sums w and row moments $\sum_v T_{sv}\eta_v = w_s\xi_s$.

Step 4: verify the observed-token conditional law. For an observed right token t, first choose one of its atoms s with probability w_s/b_t, then sample the transfer row $T_{s\cdot}/w_s$, and return the destination atom's score index $J = j(v)$. Its conditional mean is

$$\mathbb{E}[A_J \mid t] = \frac{1}{b_t} \sum_{s:t(s)=t} \sum_v T_{sv}\eta_v = \frac{1}{b_t} \sum_{s:t(s)=t} w_s f_t = f_t. \tag{93}$$

Unconditionally, the source atom has mass w_s. The destination atom thus has mass $\sum_s T_{sv} = w_v$, and

$$\mathbb{P}(J = j) = \sum_{v:j(v)=j} w_v = \pi_j. \tag{94}$$

This proves both required properties of the same kernel. Independence from the left representative follows because this draw uses only the right token and fresh independent bits. The selection rule preceding the lemma now yields left output mass $a_i x_i = l_i$ and right output mass $b_t f_t = r_t$.

Step 5: count the representation and handle ties. Sorting takes $O(m_{\mathrm{LR}} \log(m_{\mathrm{LR}} + 1))$ work. The endpoint overlay and cumulative discrepancy matching each create linearly many records, and Lemma 9 gives linearly many edges. Store CDFs of the unnormalized atom and incident-edge masses. Two binary searches and the switching coin of Equation (38) sample the kernel in $O(\log(m_{\mathrm{LR}} + 1))$ work, using linear storage overall.

Equal left ratios give zero π_j and are removed before the overlay. Equal desired means or score values are allowed in Lemma 9; any edge actually used still has a strictly positive score gap by Equation (33). Source probabilities are positive on the retained supports. Thus no sampled branch divides by zero, including deterministic representatives. □

B.8 Recursive realization and final exactness

At the root compute $\rho = \mathrm{Bal}_{c_{\mathrm{strat}}}(p|_{E_q})$. We prove the following induction invariant for every group G and every input $r \in \mathcal{B}(c_G)$: the prepared plan returns an actual candidate in G, with marginal r, using only the source draws in G and fresh verification bits.

Base case: one source. If G is a leaf with law a, its coverage is $c_G(S) = a(S)$. For a full base, singleton feasibility gives $r_i \le a_i$ for every i, while both sums equal one. Hence $r = a$. Returning that leaf's actual candidate satisfies every part of the invariant.

Inductive step: prepare feasible child targets. At an internal node, first lift its shared token, if present, into separate child labels and use Lemma 17 to obtain a feasible lifted target $\tilde{r}$. Define $\beta_L = \mathrm{Bal}_{c_L}(\tilde{r}|_L)$ and $\beta_R = \mathrm{Bal}_{c_R}(\tilde{r}|_R)$. They are full feasible child bases by Lemma 14. Lemma 16 says that $\tilde{r}$ remains feasible for independent representatives with these two marginals. Thus Lemma 18 supplies a parent selection kernel before any actual candidates are examined.

Inductive step: execute and verify the output. By the induction hypothesis, each child returns an actual candidate with its prescribed base marginal. The children use disjoint sets of independent source draws and independent verification bits, so their representatives are independent, even if some original token names coincide. The prepared two-source kernel therefore returns a lifted label with law $\tilde{r}$. Removing the copy labels adds the two masses of the shared token to r_h and leaves all other coordinates unchanged. The selected token was an actual representative and hence an actual candidate in G. Only the randomness in G has been used, completing the induction invariant.

Root correction. The root now returns $Z \sim \rho$ with $Z \in C$ almost surely. Lemma 14 gives $\mathrm{TV}(p, \rho) = D_{\mathrm{strat}}$; Lemma 6 then gives $Y \sim p$ and actual membership $1 - D_{\mathrm{strat}}$, including zero membership on rejection. This proves the probability assertions of Theorem 3 for all $n \geq 1$.

Source labels can be retained when slots are permuted. Equivalently, sorting the sampled tokens by the original interval order recovers a source-ordered sequence: stratum coordinates increase, and their token ranks are nondecreasing. Equal-token exchanges do not change that sequence. All online decisions use the prepared plan and fresh coins; no preprocessing step is conditioned on the realized pool.

C Arithmetic, Parallel, and Finite-Bit Complexity

Proof outline and quantities being counted. We first count the exact-arithmetic operations of the recursive plan from Appendix B. Its outer traversal determines child targets; each outer node also makes bottom-up principal queries. These are two distinct tree traversals. We then bound the bit length of the *actual* targets using integer witnesses, and bound the temporary quantities computed from those targets. Finally, we replace probability-coin primitives by fair-bit routines and derive bounds on their running-time tails. The integer witnesses are used in the analysis; they do not change the sampler or its child targets.

C.1 Computational model and parallel primitives

The arithmetic model stores exact rational scalars and integer indices. An arithmetic operation or comparison has unit cost; an exact probability coin is a primitive for the online arithmetic bound. For parallel preprocessing we use concurrent-read, exclusive-write (CREW) memory, and count total work, critical-path span, and peak live storage separately. Let

$$M = V + n, \qquad L = \lceil \log_2(M+1) \rceil, \qquad h = \lceil \log_2(n+1) \rceil. \tag{95}$$

The implementation uses arrays, offsets, and deterministic sort/scan indexing. Table 3 records the primitives used in the recursive construction.

Table 3: Exact-arithmetic CREW primitives on m records. Each primitive uses $O(m)$ live storage with local double buffering.

Primitive	Work	Span
Stable merge, prefix scan, compaction	$O(m)$	$O(\log(m+1))$
Mergesort, upper hull, canonical deduplication	$O(m \log(m+1))$	$O(\log^2(m+1))$

Stable merge and overlays. For two sorted arrays, stable co-ranking identifies the input prefixes associated with any output prefix (Siebert & Träff, 2013). Divide the output into blocks of size $\Theta(\log(m+1))$. Compute each block's endpoint co-ranks in parallel, then merge within blocks. There are $O(m/\log(m+1))$ endpoint searches, each with logarithmic cost; total work is linear and

span logarithmic. Ties are consistently resolved in favor of one source array. A balanced mergesort therefore has $O(m \log(m+1))$ work and $O(\log^2(m+1))$ span. The same merge followed by prefix scans constructs the cumulative-endpoint overlays in Lemmas 9 and 18. In the product merge, source-origin prefix counts supply every pair of child indices without a serial pointer walk.

Upper hulls. First sort by horizontal coordinate, retain the highest vertical coordinate at a repeated horizontal coordinate, and remove duplicate points. Build hulls by a balanced merge of horizontally separated chains. If A, B are two such chains and $F_A(t) = \max_{(x,y)\in A}(tx + y)$, then

$$[F_B(t_2) - F_A(t_2)] - [F_B(t_1) - F_A(t_1)] \geq (t_2 - t_1)(\min x(B) - \max x(A)) > 0 \qquad (96)$$

for $t_2 > t_1$. Thus the supporting-chain switch is monotone. To find its bridge, evaluate the support-function difference at the ordered rational edge breakpoints. All breakpoint queries can be performed by merging the two breakpoint arrays, using source-prefix ranks and scans to identify support points. The sign transition locates the bridge; if it lies between breakpoints the two active vertices are its endpoints. At equality keep the leftmost support on the left chain and the rightmost on the right, discarding collinear interior points. This is a linear-work, logarithmic-span hull merge using only rational comparisons. Balanced merging and the initial sort give Table 3. Single-point chains and zero-failure endpoints follow the same tie conventions.

References and zero weights. Canonical internal keys contain a constant number of child IDs and a local inclusion bit. Sorting and scanning assign equal keys equal IDs with exclusive writes to predetermined output offsets. Boundary masks remain separate from these internal keys. Expand all adjacent chain differences in parallel, skipping equal subtrees; prefix sums allocate child work queues. Because the recovered sets are nested, a visited unequal subtree contains a token added in that adjacent difference. Charge the visit to such a token at that tree depth. A token is added only once and has at most $O(\log(n_G + 1))$ ancestors, so the total expansion size is $O(M_G \log(n_G + 1))$. Breadth-first queues bound its span by $O(\log(n_G + 1) \log(M_G + 1))$.

For zero-weight extension, let $b_{r,0} = 1 - a_r(W_+)$ and let $b_{r,j}$ be the failure in row r after adding the first j zero-weight tokens. A row changes only at its token incidences. Segmented prefix sums over events in each row compute all these old and new factors. If both factors are positive, record the ratio $b_{\text{new}}/b_{\text{old}}$. If a positive factor becomes zero, record a zero-count increment and use ratio one; an already zero factor stays zero. Initially zero factors are counted separately.

Sort the events by token order and scan their ratios and zero-count increments. At the last event of token z_j, the resulting failure is

$$\varphi_j = \begin{cases} 0, & \text{if at least one row factor is zero,} \\ \left(\prod_r b_{r,0}\right)\left(\prod_{\text{events through } j} b_{\text{new}}/b_{\text{old}}\right), & \text{otherwise.} \end{cases} \qquad (97)$$

The second branch uses only positive denominators. The greedy increment is $\beta_{z_j} = \varphi_{j-1} - \varphi_j$. There are $O(M_G + n_G)$ events and initial factors. Sorting and scans give the work and span in Table 3, including cases where the initial coverage is already one.

C.2 The two tree traversals

Inner traversal: one principal query. A principal query on a group with M_G incidences and n_G sources uses a bottom-up tree of the primitives above. At a fixed inner depth, the source groups partition the group's rows. Their incidence counts sum to M_G, and each group stores only a constant number of boundary masks. A curve has at most one exposed vertex per free positive-weight token plus its initial vertex, so the total record size at that depth is $O(M_G)$.

Each node's local sorts, merges, hulls, and canonical indexing cost $O(m \log(m+1))$ work and $O(\log^2(m+1))$ span on its m records. Summing work across the depth costs $O(M_G \log(M_G + 1))$; taking the maximum local span costs $O(\log^2(M_G + 1))$. There are $O(\log(n_G + 1))$ dependent inner depths. Including the chain recovery and zero extension above therefore gives

$$O(M_G \log(M_G + 1) \log(n_G + 1)), \qquad O(\log^2(M_G + 1) \log(n_G + 1)). \qquad (98)$$

Outer traversal: prepare the actual child targets. At an outer realization node, a constant number of conditional queries determines the shared split and child bases, after which the two-source kernel is constructed. The constants come from the two shared-token states and the two children. Only after these operations are the actual child targets available for recursive preprocessing. Thus queries at the next outer depth depend on the preceding depth, even though their own inner traversals run bottom-up.

At outer depth d, groups have disjoint source rows and their incidence counts sum to $O(M)$. Each inner query has remaining height $O(h-d)$. Summing Equation (98), including the root query and the smaller kernel costs, gives

$$W_{\mathrm{pre}} = O\left(ML\sum_{d=0}^{h-1}(h-d)\right) = O(MLh^2), \tag{99}$$

$$T_{\infty,\mathrm{pre}} = O\left(L^2\sum_{d=0}^{h-1}(h-d)\right) = O(L^2h^2). \tag{100}$$

Here $\sum_{d=0}^{h-1}(h-d) = h(h+1)/2$. The inner height contributes one factor of h, and the sequential dependence among outer target levels contributes the other. The smaller two-source costs sum to $O(MLh)$ work and $O(L^2h)$ span and are absorbed. The root principal query is absorbed as well.

Peak live storage under the level schedule. At outer depth d, a query may retain references from its remaining inner levels. Across all active groups these query arenas occupy $O(M(h-d+1))$ scalars and indices. Each already prepared outer depth saves only linearly many target and kernel records, giving $O(M(d+1))$ saved objects. Complete a depth, save its kernels and child targets, and release its query arenas before starting the next depth. Local double buffering multiplies these sizes only by a constant. At every barrier the total is bounded by

$$O(M(h-d+1)) + O(M(d+1)) = O(M(h+2)) = O(Mh). \tag{101}$$

Online work and span. During execution there are at most $n-1$ internal selections. Each performs a constant number of CDF searches and switching operations, hence $O(L)$ work and span. Summing over nodes gives $O(nL)$ work. Along any root-to-leaf dependency path there are $O(h)$ selections, giving $O(hL)$ span; nodes on separate branches can run concurrently. The final correction and residual categorical draw add $O(L)$ to each bound. Retaining source labels avoids sorting the actual pool; an optional rank sort costs $O(n\log(n+1))$ work and $O(\log^2(n+1))$ span, still within these bounds. This proves the arithmetic and parallel parts of Theorem 3.

C.3 Input encoding and the IID bit bound

Encode every probability as a binary nonnegative numerator and a positive denominator, and normalize after each rational scalar operation. Let B_{in} denote total input bit length. For the IID bound, take D_p, D_q to be common multiples of the *encoded* denominators of p, q, respectively, including denominators of unreduced input fields. This also bounds the cost of normalizing those fields. With reduced input, their least common multiples suffice. Define

$$B_A = O\big(\log(2D_p) + n\log(2D_q) + \log(V+n+1)\big). \tag{102}$$

The prefix probabilities lie on the corresponding denominator grids, and the powers Q_j^n lie on $D_q^{-n}\mathbb{Z}$. All f, h, R, ρ, v, δ and cumulative transfer masses consequently lie on a common grid whose denominator divides $D_pD_q^n$. Their magnitudes and all ratio comparisons have $O(B_A)$-bit numerators and denominators. The sampler stores cumulative *unnormalized* transfer masses, and each switching probability is a constant-depth rational expression in these quantities. It does not accumulate the separately divided entries $e/(y_j - y_i)$ to obtain a diagonal.

Let $\mathsf{Mult}(b)$ be the cost of multiplying b-bit integers; normalized rational operations can be implemented in $O(\mathsf{Mult}(b)\log(b+1))$ bit work using fast integer arithmetic and gcd (Brent & Zimmermann, 2010). Replacing the arithmetic operations counted in Appendix A.4 yields preprocessing work

$$O\left(B_{\mathrm{in}} + V[\log(V+1) + \log(n+1)]\,\mathsf{Mult}(B_A)\log(B_A+1)\right). \tag{103}$$

The bound accounts for the given input representation; it does not assume that unrelated rational denominators share a small common grid. Rational comparisons use cross-products and do not require real-number roots.

C.4 INTEGER WITNESSES FOR RECURSIVE TARGET SIZE

For the recursive verifier, normalize the input first, and let D_p, D_q be common denominators of the reduced vectors. Put $b_p = \lceil \log_2(D_p + 1) \rceil$, $b_q = \lceil \log_2(D_q + 1) \rceil$, and let b_{enc} bound the length of every original input field. Each source mass a_{ri} has denominator dividing D_q. A coverage value for g sources therefore lies on $D_q^{-g}\mathbb{Z}$.

The proof tracks, but does not compute, nonnegative integer weights U_G such that every actual recursive target is $\mathrm{Bal}_{c_G}(U_G)$. We choose witnesses with positive support equal to the positive support of that base. Three properties keep their lengths additive along a source-tree path.

Lemma 19 (Subgroup reweighting)**.** *For independent pools on disjoint labels, suppose $r = \mathrm{Bal}_c(u)$ and r, u have the same positive support. With the same fixed-order zero extension,*

$$\mathrm{Bal}_{c_L}(r|_L) = \mathrm{Bal}_{c_L}(u|_L), \qquad \mathrm{Bal}_{c_R}(r|_R) = \mathrm{Bal}_{c_R}(u|_R). \tag{104}$$

Proof. **Step 1: identify the child blocks inside the parent chain.** It suffices to prove the left identity. Write $L_0 \subset \cdots \subset L_s$ for the left positive-weight principal chain under $u|_L$, with block slopes α_j and blocks $J_j = L_j \setminus L_{j-1}$. Write (A_t, B_t) for the left and right portions of successive parent principal sets, and ℓ_t for their parent block slopes.

For fixed B, the parent minimization has the form

$$c_R(B) + (1 - c_R(B))c_L(A) - \lambda[u(A) + u(B)]. \tag{105}$$

If $1 - c_R(B) > 0$, its minimization over A is a child principal query at parameter $\lambda/(1 - c_R(B))$. If that failure is zero and $\lambda > 0$, all positive-weight left tokens are included. Hence parent endpoints may be chosen to project to child endpoints. With the maximal endpoint convention at ties, each child block enters as a whole at a parent step $t(j)$, where $t(j)$ is nondecreasing in j.

For $i \in J_j$, the old child base and the parent base satisfy

$$[\mathrm{Bal}_{c_L}(u|_L)]_i = \alpha_j u_i, \qquad r_i = \ell_{t(j)} u_i. \tag{106}$$

The assumption $\mathrm{supp}(r) = \mathrm{supp}(u)$ ensures $\ell_{t(j)} > 0$. Thus the ratio of the old child base to the new weight r_i is $\alpha_j/\ell_{t(j)}$.

Step 2: compare consecutive child ratios across parent steps. If $t(j) = t(j+1)$, the ratios are nondecreasing because $\alpha_j \le \alpha_{j+1}$. Otherwise put $t = t(j) < t' = t(j+1)$ and define

$$b_{\mathrm{old}} = 1 - c_R(B_t), \qquad b_{\mathrm{new}} = 1 - c_R(B_{t'-1}). \tag{107}$$

These are the other child's failures *after* the earlier parent step and *before* the later one. Nesting gives $b_{\mathrm{old}} \ge b_{\mathrm{new}}$.

At parameter ℓ_t, the endpoint after step t is optimal. Compare it with the same set with the last entering left block J_j removed. The objective difference is

$$b_{\mathrm{old}}[c_L(L_j) - c_L(L_{j-1})] - \ell_t u(J_j) \le 0. \tag{108}$$

At parameter $\ell_{t'}$, the endpoint before step t' is optimal. Compare it with the same set with J_{j+1} added; this gives

$$b_{\mathrm{new}}[c_L(L_{j+1}) - c_L(L_j)] - \ell_{t'} u(J_{j+1}) \ge 0. \tag{109}$$

Divide by the respective positive block weights. Since their coverage-to-weight ratios are α_j, α_{j+1}, we obtain

$$\ell_{t(j)} \ge b_{\mathrm{old}} \alpha_j, \qquad \ell_{t(j+1)} \le b_{\mathrm{new}} \alpha_{j+1}. \tag{110}$$

Here $b_{\mathrm{new}} > 0$: otherwise the second bound would give $\ell_{t'} \le 0$, contradicting positive parent mass on J_{j+1}. Consequently $b_{\mathrm{old}} > 0$, and all divisions in

$$\frac{\alpha_j}{\ell_{t(j)}} \le \frac{1}{b_{\mathrm{old}}} \le \frac{1}{b_{\mathrm{new}}} \le \frac{\alpha_{j+1}}{\ell_{t(j+1)}} \tag{111}$$

are valid. This proves monotonicity of all consecutive child ratios.

Step 3: identify the resulting balanced base. The old child base is feasible independently of which weights describe it, and its child chain remains tight. Steps 1–2 show that it is proportional to $r|_L$ on that chain with nondecreasing ratios. Since $r|_L$ and $u|_L$ have identical positive supports, the remaining fixed-order zero extension starts from the same set and has the same increments. Lemma 15 therefore identifies this base as $\mathrm{Bal}_{c_L}(r|_L)$. If both restricted weight vectors are zero, the identical greedy extension proves the equality directly. The right identity follows by exchanging the children. □

Lemma 20 (Finite integer witness for zero extension)**.** *Suppose U is a nonnegative integer weight vector on a g-source group. Its fixed-order zero extension can be represented by nonnegative integer weights with exactly the base's positive support, adding at most $g \log_2 D_q + z + O(1)$ bits to the largest coordinate length, where z is the number of newly positive coordinates.*

Proof. **Step 1: remove positive weights that receive zero mass.** Such coordinates lie in initial zero-slope blocks of the principal chain. Their union has coverage zero, so each is a loop: monotonicity and submodularity imply $c(S \cup \{i\}) = c(S)$ whenever $c(\{i\}) = 0$. Removing its weight therefore leaves the positive-slope chain and the base unchanged. All retained original positive weights now have positive base mass.

Step 2: give the new positive coordinates finite weights. Every original positive block has slope at most one, since its coverage increment is at most one and its integer weight is at least one. List, in the prescribed extension order, the zero-weight labels that receive positive greedy masses as $z_1, \dots, z_z$, with masses $d_j > 0$. Temporarily give z_j weight $d_j/2^j$. Its ratio of base mass to weight is exactly 2^j.

Keep the original weights on the retained coordinates. The old positive chain followed by these greedy singleton extensions is tight and has nondecreasing ratios: the old ratios are at most one, followed by $2, 4, \dots, 2^z$. A skipped zero-mass greedy coordinate has zero marginal increment at its prefix and at every later prefix by diminishing returns, so skipping it does not change later increments. The base remains feasible. Lemma 15 now proves that these finite rational weights give exactly the original base.

Step 3: clear denominators and count the added bits. Every coverage value on this group is a multiple of D_q^{-g}, so $D_q^g d_j$ is an integer. Multiplying all temporary weights by $D_q^g 2^z$ gives

$$U'_i = D_q^g 2^z U_i, \qquad U'_{z_j} = D_q^g d_j 2^{z-j}. \tag{112}$$

Common positive scaling preserves every balanced-base coordinate: it scales the weights and inversely scales their block slopes. Thus U' is an integer witness for the same base. Its positive support is exactly the base's positive support. The multiplier on an old coordinate has binary length at most $g \log_2 D_q + z + 1$; a new coordinate is at most $D_q^g 2^z$. Hence the largest coordinate length increases by at most $g \log_2 D_q + z + O(1)$, also when the original weights were all zero. Zero-mass labels remain zero. □

Lemma 21 (Shared split localizes to one principal block)**.** *Suppose a positive-mass shared token h lies in principal block J of a target $r = \mathrm{Bal}_c(U)$, where U is an integer witness with $\mathrm{supp}(U) = \mathrm{supp}(r)$. Write $\Delta c = a/D_q^g$ for that block's positive coverage increment and $\ell = \Delta c/U(J)$. If the feasible split t from Lemma 17 is positive, then*

$$\frac{t}{\ell} = U(T) + \frac{bU(J)}{a} \tag{113}$$

for some subset T of the block and integer b. Scaling all split witness coordinates by $a \le D_q^g$ gives an integer witness for the lifted target, adding $g \log_2 D_q + O(1)$ bits.

Proof. **Step 1: localize the optimizing cut with two explicit inequalities.** Let A, B be the principal sets immediately before and after block J, lifted so that A contains neither copy of h and B contains both. They are tight under the all-right reference target r^0. Put $g_0(S) = c_{LR}(S) - r^0(S)$. This function is submodular, and $g_0(A) = g_0(B) = 0$.

Start with any only-right cut S. Submodularity for S, A gives

$$g_0(S \cup A) \le g_0(S) + g_0(A) - g_0(S \cap A) \le g_0(S). \tag{114}$$

Indeed, $S \cap A$ contains neither copy and is a compatible original-token cut, so its slack is nonnegative. Next apply submodularity to $S \cup A, B$:

$$g_0((S \cup A) \cap B) \le g_0(S \cup A) + g_0(B) - g_0((S \cup A) \cup B) \le g_0(S \cup A). \tag{115}$$

Here the final union contains both copies and is again a compatible cut with nonnegative slack. Combining yields

$$g_0(S \cup A) \le g_0(S), \qquad g_0((S \cup A) \cap B) \le g_0(S \cup A). \tag{116}$$

The resulting set is still only-right and lies between A and B. Since its slack did not increase, its deficit did not decrease. Thus some maximum only-right deficit is attained inside this one block. The same two comparisons also localize an only-left extremum: its intersection with A contains neither copy, and its union with B contains both.

Step 2: express the split using one block denominator. For $t > 0$, Lemma 17 says $t = C_-$. Take a localized maximizing cut S_*, and let $T \subseteq J$ be the original labels of $S_* \setminus A$, identifying its right copy with h. Block proportionality and endpoint tightness give

$$t = r^0(S_*) - c_{LR}(S_*) \tag{117}$$

$$= \ell U(T) + c_{LR}(A) - c_{LR}(S_*) \tag{118}$$

$$= \ell U(T) + \frac{b}{D_q^g}, \qquad b = D_q^g[c_{LR}(A) - c_{LR}(S_*)] \in \mathbb{Z}. \tag{119}$$

Since $\ell = a/(D_q^g U(J)) > 0$, division gives Equation (113). No sum of denominators from unrelated principal blocks is introduced.

Step 3: identify an integer witness for the lifted target. Assign weights t/ℓ and $U_h - t/\ell$ to h_L, h_R, and keep all other witness weights. Lemma 17 gives $0 \le t \le r_h = \ell U_h$, so these weights are nonnegative and sum to U_h. On every lifted original block the target remains proportional to these weights with the old slope. The old chain endpoints remain tight because each endpoint contains both copies or neither, and the lifted target is feasible by Lemma 17. Its positive support matches that of the split witness. Lemma 15 therefore identifies it as the balanced base of this witness.

Equation (113) shows that multiplying by a clears the new denominators. Since $0 < a \le D_q^g$, this adds at most $g \log_2 D_q + O(1)$ bits. For $t = 0$, the unscaled all-right witness already works; a zero-mass shared token requires no positive split. Finally, Lemma 19 identifies the sampler's actual child bases as the balanced bases of the restricted lifted witness. Lemma 20 supplies their matching-support integer witnesses if a child zero extension adds positive coordinates. □

The witness-height recurrence. Initialize the root with integer weights $D_p p_i$ on the proposal support and apply Lemma 20. Let H_j bound the largest coordinate length of a matching-support witness at the jth group of a source-tree path, whose source count is g_j. The three preceding lemmas give the following explicit charges:

1. The root zero extension adds at most $nb_q + V + 2$ bits, so $H_0 \le b_p + nb_q + V + 2$.
2. At a parent of size g_j, the shared split adds at most $g_j b_q + 1$ bits. Restricting this witness to a child does not increase coordinate lengths, and Lemma 19 identifies the child's actual base.
3. The child's zero extension adds at most $g_{j+1} b_q + V + 2$ bits. A single group has at most one label per original token; the two copies at a split enter different children. Thus at most V coordinates are newly positive in each extension.

In particular, a conservative one-level recurrence is

$$H_{j+1} \le H_j + (g_j + g_{j+1}) b_q + V + 3. \tag{120}$$

There are at most h edges on a balanced path. Since $g_j \le n/2^j + 1$, we have $\sum_j g_j \le 2n + h$. The root term n and the split/extension source counts together are at most $2\sum_j g_j$. Summing the recurrence, with slack for endpoint rounding, yields

$$H_U \le b_p + (4n + 2h) b_q + V(h + 1) + 3h + 5. \tag{121}$$

The bound is additive along the path. No step multiplies an existing coordinate length by the number of tokens. The witnesses describe the same child targets used by the recursive sampler; the sampler itself passes their reduced balanced-base coordinates.

C.5 Actual intermediate quantities and a uniform bit bound

Define

$$\Gamma = b_p + (8n+8)b_q + 2(V+1)(h+1) + 8L + 32, \tag{122}$$

$$\mathfrak{b} = 2^{16}(b_{\mathrm{enc}} + M\Gamma). \tag{123}$$

Step 1: bound the actual target coordinates. Let U be a group witness bounded by H_U, and let a principal block J have coverage increment a/D_q^g. Every positive target coordinate in that block has the representation

$$r_i = \frac{aU_i}{D_q^g U(J)}, \qquad 0 \le a \le D_q^g, \qquad U(J) \le V\,2^{H_U}. \tag{124}$$

Its numerator and denominator therefore have at most $H_U + gb_q + \lceil \log_2(V+1) \rceil + 2$ bits before reduction. This is bounded by the chosen Γ. The matching-support witness already incorporates any positive zero-extension coordinates.

For a positive shared token, Step 1 of Lemma 21 localizes either one-copy extremum to its principal block. Its value has the form

$$\frac{aU(T) + bU(J)}{D_q^g U(J)}, \qquad |b| \le D_q^g. \tag{125}$$

The same coordinate-length bound, with one extra addition bit, applies to C_+, C_- and the selected split t. A zero-mass shared token can be passed with split zero and needs no extremum query. Thus the actual targets supplied to successive recursive calls remain controlled by Γ.

Step 2: bound one principal query's temporary quantities. A query takes already constructed rational weights, each of reduced length at most Γ, on at most M labels. Let D_w be the product of those reduced denominators, used only for analysis. Then

$$\log_2 D_w \le M\Gamma, \qquad w(S) \in D_w^{-1}\mathbb{Z}, \qquad \varphi_G(S) \in D_q^{-g}\mathbb{Z}. \tag{126}$$

Every curve point is consequently on a grid with denominator dividing $D_w D_q^g$. Endpoint differences, segment ratios, τ values, and hull support comparisons are constant-depth rational expressions in these two coordinates. Their reduced lengths are bounded by a constant multiple of $M\Gamma$. This argument applies to the subset sums of the *current* query weights; it is not a recurrence that multiplies the next target's bit length by M.

For the zero-extension scans, each individual old or new row failure lies on the original D_q^{-1} grid. There are $O(M)$ incidence events. Even the uncancelled product of all positive event ratios therefore has $O(Mb_q)$ bits. Tracking zero factors separately avoids division by zero. Normalization after each scalar operation keeps all partial scan products within this bound.

Step 3: exhibit a common grid for the two-source kernel. Use the notation a_i, b_t, l_i, r_t of Appendix B.7, with combined support $m_{\mathrm{LR}} \le M$. Form an analysis denominator κ as the product of (i) the reduced denominators of all source and target coordinates and (ii) the positive reduced numerators of the source coordinates a_i, b_t. There are at most $3m_{\mathrm{LR}}$ such factors, each with at most Γ bits. Hence

$$\log_2 \kappa \le 3m_{\mathrm{LR}}\Gamma. \tag{127}$$

The ratios $x_i = l_i/a_i$ and $f_t = r_t/b_t$ have denominators dividing κ: their denominators divide a target denominator times a source numerator. The cumulative source masses A_j, score masses π_j, and their cumulative endpoints also lie on $\kappa^{-1}\mathbb{Z}$. It follows that the overlay atom lengths w_s and the values ξ_s, η_s do as well.

The discrepancy $w_s(\xi_s - \eta_s)$ lies on $\kappa^{-2}\mathbb{Z}$, as do the cumulative discrepancies, matched edge masses, and their unnormalized CDFs. All these masses have absolute value at most one. Their reduced lengths are therefore at most a constant times $\log_2 \kappa$. An atom-choice probability, an edge-choice probability, or a switching coin is a ratio of two of these bounded quantities, with positive denominator on every executed branch. This proves an $O(M\Gamma)$ bound for the complete stored kernel and its online thresholds without summing the separately divided entries of a conceptual matrix.

Step 4: cover raw operations and integer scratch space. The preceding bounds describe structured quantities and constant-depth expressions. Before normalization, rational addition and comparison use cross-products, and multiplication or division multiplies pairs of numerators and denominators. Each such operation increases the operand length by at most a constant factor. For example, the query grid has logarithmic size at most $(M+1)\Gamma$, and the kernel grid at most $3M\Gamma$; even allowing 64 times these grid sizes for unreduced expressions is below the factor 2^{16} in Equation (123). The input-field allowance b_{enc} covers normalization of unreduced input. Binary gcd, division remainders, and fair-bit comparison remainders require only constant-factor additional integer lengths within the same conservative envelope.

The child targets are the balanced-base coordinates bounded in Step 1. The larger denominators in a saved sampling kernel are never passed back as the next recursive target. Thus every scalar operation in the actual arithmetic DAG is covered by the uniform polynomial envelope $\mathfrak{b}$.

Step 5: substitute bit costs into the same arithmetic DAG. Schoolbook multiplication, restoring division, and binary gcd on $\mathfrak{b}$-bit integers give $O(\mathfrak{b}^2)$ work and span and $O(\mathfrak{b})$ scratch per normalized scalar operation. Replace each node of the arithmetic DAG by that integer routine. Summing over arithmetic nodes multiplies work by $O(\mathfrak{b}^2)$; summing along a critical path multiplies span by the same factor. A live arithmetic object or active routine uses $O(\mathfrak{b})$ bits, so the levelwise space schedule multiplies by $O(\mathfrak{b})$. Including input storage and reading gives

$$W_{\mathrm{bit,pre}} = O(B_{\mathrm{in}} + MLh^2\mathfrak{b}^2), \tag{128}$$

$$T_{\infty,\mathrm{bit,pre}} = O(\log(B_{\mathrm{in}}+2) + L^2h^2\mathfrak{b}^2), \tag{129}$$

$$S_{\mathrm{bit,peak}} = O(B_{\mathrm{in}} + Mh\mathfrak{b}). \tag{130}$$

Input normalization is covered by b_{enc} and the same integer-operation bounds. Thus the exact construction has polynomial bit complexity in the input length, V, and n. The polylogarithmic arithmetic span and the polynomial bit span are separate guarantees.

C.6 Exact rational coins from fair random bits

One coin: exactness, termination, and bit cost. For $0 < a < b$, compare a fresh uniform binary expansion U with the binary expansion of a/b. Generate digits until they differ and return whether $U < a/b$. Starting with remainder $r_0 = a$, the next threshold digit and remainder are computed by

$$d_s = \left\lfloor \frac{2r_{s-1}}{b} \right\rfloor \in \{0,1\}, \qquad r_s = 2r_{s-1} - d_s b, \qquad 0 \le r_s < b. \tag{131}$$

Doubling, comparing, and subtracting cost $O(\mathfrak{b})$ bit operations per digit. After t coincident digits, the unresolved values of U lie in a single dyadic interval of length 2^{-t}. Conditional on any prior computation history $\mathcal{H}$, the fresh draw U remains uniform. Thus the number B_{coin} of fair bits used satisfies

$$\mathbb{P}(B_{\mathrm{coin}} > t \mid \text{history}) \le 2^{-t}, \qquad \mathbb{E}[B_{\mathrm{coin}} \mid \text{history}] \le 2. \tag{132}$$

The expectation bound follows by summing the tail: $\mathbb{E}[B_{\mathrm{coin}} \mid \mathcal{H}] = \sum_{t\ge 0} \mathbb{P}(B_{\mathrm{coin}} > t \mid \mathcal{H}) \le 2$. Letting $t \to \infty$ proves almost-sure termination. The event of an infinite tie has probability zero, so the output is exactly Bernoulli(a/b). Probabilities zero or one need no randomness; a dyadic threshold can use its finite binary comparison.

A categorical CDF search is a balanced decision tree. Within a current interval of positive mass, its next branch probability is the mass of the left subinterval divided by the current interval mass. Zero-mass branches are skipped deterministically. The tree has logarithmic depth, and forming its rational thresholds is included in the online scalar-operation count.

Many adaptive coins: derive a uniform exponential bound. The conditional tail above stochastically bounds B_{coin} by a geometric variable G with $\mathbb{P}(G = s) = 2^{-s}$ for $s \ge 1$. Choose $\theta = \log(4/3)$; then

$$\mathbb{E}[e^{\theta B_{\mathrm{coin}}} \mid \mathcal{H}] \le \sum_{s\ge 1} 2^{-s}(4/3)^s = \sum_{s\ge 1}(2/3)^s = 2. \tag{133}$$

There is a deterministic upper bound $N = CnL$ on the number of coin calls per output, for an absolute constant C. Let B_j be the bit count of call j, padding unused calls by zero. In any dependency-respecting order, conditional expectation and Equation (133) imply inductively

$$\mathbb{E}\exp\left(\theta\sum_{j=1}^{N} B_j\right) \le 2^N. \tag{134}$$

This uses a conditional bound at each invocation, not independence of the random running times. Markov's inequality now gives, for every $t \ge 0$,

$$\mathbb{P}\left(\sum_{j=1}^{N} B_j > \frac{N\log 2 + t}{\theta}\right) \le e^{-t}. \tag{135}$$

Summing conditional means also gives $\mathbb{E}\sum_j B_j \le 2N$. With $t = \log(1/\eta)$, these prove

$$\mathbb{E}B_{\mathrm{total}} = O(nL), \qquad B_{\mathrm{total}} = O(nL + \log(1/\eta)) \tag{136}$$

with probability at least $1 - \eta$.

Work: include rational threshold preparation. Outside bit comparisons, online execution has $O(nL)$ rational operations, each costing $O(\mathfrak{b}^2)$. Each generated fair bit costs $O(\mathfrak{b})$ for the remainder update. Thus

$$W_{\mathrm{on,bit}} \le C_1 nL\mathfrak{b}^2 + C_2\mathfrak{b}B_{\mathrm{total}}. \tag{137}$$

The mean and tail above yield expected work $O(nL\mathfrak{b}^2)$ and, with probability at least $1 - \eta$, work $O(nL\mathfrak{b}^2 + \mathfrak{b}\log(1/\eta))$.

Span: bound all dependency paths. Each source-tree path contains $O(hL)$ coin calls, including the final correction, and $O(hL)$ other rational operations. There are at most n leaf-to-root paths. Apply Equation (135) to each fixed path, conditioning on the information available at each of its invocations. A union bound with $t = \log(n/\eta)$ gives

$$\max_{\text{paths }\pi}\sum_{j\in\pi} B_j = O(hL + \log n + \log(1/\eta)) = O(hL + \log(1/\eta)) \tag{138}$$

with probability at least $1 - \eta$. Within a node, the constant number of searches may be executed sequentially, so these tree paths cover the online dependencies. Multiplying bit comparisons by $O(\mathfrak{b})$ and adding the rational-operation path costs gives span $O(hL\mathfrak{b}^2 + \mathfrak{b}\log(1/\eta))$. Integrating the exponential tail gives expected span $O(hL\mathfrak{b}^2)$. Work and span bounds can hold simultaneously by allocating $\eta/2$ to each event.

The parameter η controls running time only. The algorithm never truncates a coin, so these high-probability bounds do not introduce any error into the output distribution.

D Marginal Headroom and Continuation-Value Guarantees

The first proof constructs the target-aware reference used in the empirical frontier. The second characterizes nonnegative weighted membership by a finite flow problem. The final argument applies that characterization at each state of a finite-horizon continuation process.

D.1 The target-aware marginal oracle

Step 1: bound each token's successful mass. We prove Proposition 4 constructively. For any proposal coupling whose individual slots have marginal q,

$$\mathbb{P}(Y = i,\ i \in C) \le \min\{p_i, \mathbb{P}(i \in C)\} \le \min\{p_i, nq_i\}. \tag{139}$$

The first bound follows because successful mass at i is contained in both $\{Y = i\}$ and $\{i \in C\}$. For the second, the union bound gives $\mathbb{P}(i \in C) \le \sum_{r=1}^{n}\mathbb{P}(X_r = i) = nq_i$. Membership events for different output tokens are disjoint; summing therefore proves the oracle upper bound.

Step 2: construct a joint law attaining the bound. Put $s_i = \min(p_i, nq_i)$, $a = \sum_i s_i$, and $r_i = nq_i - s_i$. These quantities obey

$$0 \le a \le 1, \qquad r_i \ge 0, \qquad \sum_i r_i = n - a. \tag{140}$$

Define a success event labeled i of mass s_i and a failure event labeled i of mass $p_i - s_i$. Their masses are nonnegative and sum to $\sum_i p_i = 1$. Set $Y = i$ on either event. This already gives $\mathbb{P}(Y = i) = s_i + (p_i - s_i) = p_i$.

On a success event labeled i, place one mandatory copy of i into the pool, leaving $n - 1$ slots. On a failure event, leave all n slots unfilled. For $n - a > 0$, fill each remaining slot independently from $r/(n - a)$ and apply an independent uniform permutation to the completed pool.

Step 3: verify every proposal-slot marginal. Let N_i be the count of token i before permutation. The mandatory-copy contribution to $\mathbb{E}N_i$ is s_i. Conditional on an event with m unfilled slots, their expected contribution is $mr_i/(n-a)$. Since the expected number of such slots is $a(n-1)+(1-a)n = n - a$, total expectation gives

$$s_i + (n - a)\frac{r_i}{n - a} = nq_i. \tag{141}$$

Conditional on the unpermuted multiset, a uniform permutation puts token i in any specified slot with probability N_i/n. Averaging gives $\mathbb{P}(X_r = i) = \mathbb{E}N_i/n = q_i$ for each ordered slot r.

Step 4: check actual membership and the boundary case. Every success event contains its mandatory output token. If a failure event labeled i has positive mass, then $p_i > s_i$, forcing $s_i = nq_i$ and $r_i = 0$. All n slots on that failure event are filled from a law assigning zero mass to i, so its membership probability is zero. Actual membership is therefore exactly the success mass $a = \sum_i \min(p_i, nq_i)$.

If $n - a = 0$, then $n = 1$ and $a = 1$. In that case $s = p = q$, every event is a success, and the sole mandatory copy fills the pool without a residual draw. This handles the only case with a zero denominator. The construction explicitly uses p when generating the pool and thus establishes the target-aware Marginal Oracle.

D.2 Weighted membership as a polytope problem

For nonnegative token values $v \in \mathbb{R}_+^E$, define

$$\Phi_{\mathcal{Q}}(p, v) = \max_{(X,Y):\ X\sim\mathcal{Q},\ Y\sim p} \mathbb{E}[v_Y \mathbf{1}\{Y \in C(X)\}]. \tag{142}$$

Let

$$\mathcal{P}_{\mathcal{Q}}(p) = \{x \in \mathbb{R}_+^E : x_i \le p_i,\ x(S) \le c_{\mathcal{Q}}(S) \text{ for all } S\}. \tag{143}$$

Proposition 22 (Weighted coverage dominance)**.** *For every $v \ge 0$, $\Phi_{\mathcal{Q}}(p, v) = \max_{x\in\mathcal{P}_{\mathcal{Q}}(p)} v^\top x$. If $c_{\mathcal{Q}_1}(S) \ge c_{\mathcal{Q}_0}(S)$ for all S, then $\Phi_{\mathcal{Q}_1}(p, v) \ge \Phi_{\mathcal{Q}_0}(p, v)$ for all $p, v \ge 0$.*

Proof. **Step 1: map a full coupling to a feasible vector.** For any correct coupling, define $x_i = \mathbb{P}(Y = i, Y \in C)$. Then $0 \le x_i \le p_i$. For any set S, successful output in S requires the pool to hit S, so

$$x(S) = \mathbb{P}(Y \in S,\, Y \in C) \le c_{\mathcal{Q}}(S). \tag{144}$$

Thus $x \in \mathcal{P}_{\mathcal{Q}}(p)$ and the coupling's weighted value is $v^\top x$. Maximizing gives $\Phi_{\mathcal{Q}}(p, v) \le \max_{x\in\mathcal{P}_{\mathcal{Q}}(p)} v^\top x$.

Step 2: realize a feasible vector as a membership subflow. Fix $x \in \mathcal{P}_{\mathcal{Q}}(p)$ and put $m = x(E)$. In the network of Lemma 5, retain tuple supplies $\mathcal{Q}$ and replace token sink capacities by x_i. A cut indexed by token set S has capacity

$$c_{\mathcal{Q}}(S) + x(E \setminus S) = m + c_{\mathcal{Q}}(S) - x(S) \ge m. \tag{145}$$

Total sink capacity is m, so max-flow/min-cut supplies a flow of value m, saturating every token demand x_i. Its row masses do not exceed $\mathcal{Q}$, and its column masses do not exceed p.

Step 3: complete the coupling and compare weighted values. The remaining row supplies and target column demands both sum to $1 - m$. If $m < 1$, fill them by their normalized product, as in Lemma 5; if $m = 1$, they are already filled. This completion preserves the routed membership masses and may add more. Since $v_i \geq 0$, its weighted membership is at least $v^\top x$. Hence $\Phi_{\mathcal{Q}}(p, v) \geq v^\top x$ for every feasible x, proving the reverse inequality. The finite polytope is compact, so its maximum is attained.

Step 4: apply coverage dominance. If $c_{\mathcal{Q}_1} \geq c_{\mathcal{Q}_0}$ pointwise, every subset constraint defining $\mathcal{P}_{\mathcal{Q}_0}(p)$ implies the corresponding one for $\mathcal{P}_{\mathcal{Q}_1}(p)$. Therefore

$$\mathcal{P}_{\mathcal{Q}_0}(p) \subseteq \mathcal{P}_{\mathcal{Q}_1}(p), \tag{146}$$

and maximizing the same linear objective over these two sets proves the claimed ordering. □

D.3 Local continuation policies

Conditional transport also provides a framework for composing verification across candidate trees (Weng et al., 2026). Here we study how coverage dominance over all subsets orders optimal continuation values under a specified information model.

Consider a finite-horizon state space with target next-token law p_s, nonnegative reward $r_s(i)$, and a common transition $T(s, i)$. At each reached state s, method m draws a fresh candidate pool with conditional law $\mathcal{Q}_{m,s}$. Its verifier may depend on the current state and pool, and continuation ends on the first membership failure. Future random draws are unavailable to that verifier. The state includes any past information used by the policy, so this conditional description applies at each invocation.

With a horizon of d successful continuations remaining, the optimal value satisfies

$$V_{m,0}(s) = 0, \qquad V_{m,d+1}(s) = \Phi_{\mathcal{Q}_{m,s}}\left(p_s, \left(r_s(i) + V_{m,d}(T(s,i))\right)_{i \in E}\right). \tag{147}$$

Why this recursion is the optimal value. Condition on the current output i and its membership event. Failure contributes zero and terminates the continuation. Success contributes $r_s(i)$ plus the value of the common successor state with one fewer continuation remaining. By the information assumptions, the current verifier cannot condition its choice on future source draws. Thus its optimization at this state is exactly the weighted one-step coupling in Equation (147).

For the upper bound, any continuation policy has successor value at most $V_{m,d}(T(s, i))$, so optimizing its current coupling gives at most the right-hand side. For attainment, choose a maximizing current coupling and then an optimal successor policy after each successful output. The finite-horizon induction starts with value zero, and the finite coupling problem attains its maximum at each step. Hence equality holds.

Dominance by induction on the remaining horizon. Suppose the Stratified law dominates the IID law in coverage at every state. At horizon zero both values are zero. Assume $V_{\mathrm{strat},d} \geq V_{\mathrm{iid},d}$ and define the two continuation vectors $v_{m,i} = r_s(i) + V_{m,d}(T(s, i))$. They are nonnegative and satisfy $v_{\mathrm{strat}} \geq v_{\mathrm{iid}}$ coordinatewise. Increasing a nonnegative reward coordinate cannot decrease the maximum coupling value. Therefore

$$\begin{aligned} V_{\mathrm{strat},d+1}(s) &= \Phi_{\mathcal{Q}_{\mathrm{strat},s}}(p_s, v_{\mathrm{strat}}) && (148)\\ &\geq \Phi_{\mathcal{Q}_{\mathrm{strat},s}}(p_s, v_{\mathrm{iid}}) && (149)\\ &\geq \Phi_{\mathcal{Q}_{\mathrm{iid},s}}(p_s, v_{\mathrm{iid}}) = V_{\mathrm{iid},d+1}(s). && (150) \end{aligned}$$

The first inequality is reward monotonicity; the second is Proposition 22 with the *same* continuation vector. This completes the induction and proves

$$V_{\mathrm{strat},d}(s) \geq V_{\mathrm{iid},d}(s) \quad \text{for every } s, d. \tag{151}$$

The statement compares verifiers optimized for their continuation values. Theorem 3's fast construction optimizes the unit-value membership objective; Equation (147) identifies the objective for a continuation-aware extension. The depth-one latency experiment evaluates the unit-value construction.

E Experimental Protocol and Additional Results

E.1 Model states and proposal distributions

The acceptance study uses paired target and draft distributions on the same prefix. Qwen3-4B/0.6B uses cached target-only decoding for 128 output tokens, with the draft teacher-forced on those token IDs. We collect the next-token logits at positions $0, 8, \ldots, 120$. Each of the 344 prompts contributes 16 states. Both models use their non-thinking chat template and BF16 forward execution; their tokenizations of the evaluated prompts agree. The vocabulary contains 151,936 tokens.

For Qwen3-32B/1.7B, the states are positions $0, 32, 64, 96$ of the seed-zero AR trajectories on the same prompt set used by the decoding experiment. A separate causal, teacher-forced BF16 prefill extracts the four relevant logit rows from each model. These 1,376 matched states are analyzed outside the latency measurement. Gemma-2-27B-it/2B-it uses 50 prompts from each dataset and cached target-only trajectories, with the draft teacher-forced on the same tokens. The four positions $0, 32, 64, 96$ give 600 states. Gemma uses its own chat template, BF16 weights, and attention with its prescribed softcapping and sliding-window behavior; its vocabulary has 256,000 tokens. The three model-pair results describe their respective collected state distributions.

Convert each saved logit vector to FP64, divide by the specified temperature, and apply a full-vocabulary softmax. For finite K, select the largest K draft probabilities, break probability ties stably, and renormalize the selected draft mass. Positive-q tokens retain increasing token-ID order for the Stratified interval construction. The target distribution retains the full vocabulary. The acceptance evaluator may combine omitted target coordinates into a single zero-q bin: no candidate can equal such a coordinate, so their total mass suffices for membership and the deficit. The output sampler retains the original target residual over those tokens.

The default state comparison is $n = 4$, $K = 100$, and temperature 1. The Qwen3-4B sweeps use $n \in \{1, 2, 3, 4, 6, 8, 10\}$ at $K = 100$ and $K \in \{50, 100, 200, 500, 1000, \text{full}\}$ at $n = 4$, on the same 5,504 states. EOS is ignored during the fixed-length state-collection trajectories. The datasets supply varied coding, mathematics, and dialogue prefixes; the outcome measured here is exact-sampling acceptance, rather than task accuracy.

E.2 Acceptance metrics and numerical checks

Sequential IID is evaluated with its deterministic residual-chain membership calculation. Optimal IID is evaluated by Equation (4); its explicit score construction is checked separately through integrated finite couplings. Stratified uses the full construction of its fixed-law optimal plan. The oracle is Equation (13), using the effective draft distribution of each setting. All reported main and sweep constructions at temperature 1 satisfy the numerical feasibility and acceptance-ordering checks at tolerance 2×10^{-11}.

The exact-rational theorems specify the probability algorithms. Language-model experiments use BF16 forward execution and FP64 probability calculations and sampling. To validate the numerical realization, 45 finite cases compare the constructed Stratified coupling to independent rational-flow optima; their largest output-marginal L_1 discrepancy is 3.00×10^{-15} and largest membership discrepancy is 1.55×10^{-15}. Another 27 cases integrate Optimal IID's finite-score and correction branches, with maximum marginal discrepancy 2.72×10^{-15} and membership discrepancy 4.66×10^{-15}. These checks cover actual output membership, including the residual branch, using the same numerical tolerance of 2×10^{-11}. They complement the general proofs rather than estimating acceptance through Monte Carlo trials.

E.3 Aggregation and paired uncertainty

Let z_{dju} be a statistic at state u of prompt j in dataset d, and let N_{dj} be that prompt's collected state count. Dataset means weight collected states equally, and the macro mean weights the three datasets equally:

$$\bar{z}_d = \frac{\sum_j \sum_u z_{dju}}{\sum_j N_{dj}}, \qquad \bar{z} = \frac{1}{3} \sum_d \bar{z}_d. \tag{152}$$

Table 4: **Acceptance and headroom on each evaluated model pair and support.** Temperature one, $n = 4$, equal dataset weights. Membership and captured headroom are in percent; brackets give paired 95% intervals for capture. Full denotes the corresponding model vocabulary.

Target / draft	K	Sequential IID	Optimal IID	Stratified	Marginal Oracle	Headroom captured (%)
Qwen3-4B/0.6B	full	89.32	90.54	91.91	92.16	84.53 [82.40, 86.80]
Qwen3-4B/0.6B	100	89.35	90.56	91.93	92.18	84.70 [82.56, 86.94]
Qwen3-32B/1.7B	100	84.32	86.03	86.81	86.88	91.98 [88.28, 95.07]
Gemma-2-27B/2B	full	93.62	94.63	95.16	95.24	85.86 [78.62, 92.59]
Gemma-2-27B/2B	100	93.62	94.62	95.14	95.22	87.24 [80.42, 93.57]

Table 5: **Positive dependence gains across datasets.** $n = 4$, $K = 100$, temperature 1. Gain is Stratified minus Optimal IID in percentage points. Capture is the ratio of dataset-mean gaps in percent.

Target / draft	Dataset	Prompts	States	Gain [95% CI]	Capture
Qwen3-4B/0.6B	HumanEval	164	2624	1.03 [0.89, 1.18]	86.34
Qwen3-4B/0.6B	GSM8K	100	1600	1.20 [1.02, 1.38]	89.17
Qwen3-4B/0.6B	MT-Bench	80	1280	1.89 [1.60, 2.19]	81.28
Qwen3-32B/1.7B	HumanEval	164	656	1.04 [0.81, 1.29]	88.14
Qwen3-32B/1.7B	GSM8K	100	400	0.69 [0.46, 0.94]	99.69
Qwen3-32B/1.7B	MT-Bench	80	320	0.63 [0.39, 0.90]	90.80
Gemma-2-27B/2B	HumanEval	50	200	0.35 [0.13, 0.62]	97.06
Gemma-2-27B/2B	GSM8K	50	200	0.24 [0.07, 0.46]	99.97
Gemma-2-27B/2B	MT-Bench	50	200	0.99 [0.63, 1.40]	81.76

Resample prompts with replacement independently within each dataset, preserving all selected states of a sampled prompt and all methods at each state. Each resampled dataset mean is its total statistic sum divided by its total sampled state count. Thus the same state-weighted estimand is preserved even for unequal prompt-cluster sizes. Use 10,000 replicates with random seeds $190929 + j$ for dataset index $j = 0, 1, 2$ in HumanEval/GSM8K/MT-Bench order. Report the empirical 2.5th and 97.5th percentiles.

For each replicate, compute the difference between Stratified and Optimal IID before aggregation. For headroom capture, aggregate the gain and oracle gap with the same weights and then divide, as in Equation (14). When the oracle gap is unresolved at the numerical tolerance of 2×10^{-11}, the capture fraction is undefined. In particular it is undefined at $n = 1$; Figure 3 leaves that point empty. The intervals describe prompt variability in the evaluated setting. Model pairs are analyzed separately, and no cross-pair average is formed.

E.4 Full-support and per-dataset results

Table 4 includes the full-vocabulary confirmations. On Qwen3-4B/0.6B, full-support Stratified captures 84.53% of the dependence headroom, compared with 84.70% at $K = 100$. Gemma captures 85.86% with full support and 87.24% at $K = 100$. Table 5 shows each dataset within the common $K = 100$ setting. All nine paired gain intervals lie above zero.

E.5 Temperature sensitivity at $K = 100$

This study uses 384 prespecified Qwen3-4B/0.6B states and recomputes both distributions from the same saved logits at each $T \in \{0.6, 0.8, 1.0\}$, with $K = 100$ and $n = 4$. All three settings contain all 384 states. The gains over optimal IID are 1.18 pp [0.82, 1.58], 1.41 pp [1.01, 1.85], and 1.53 pp [1.12, 1.98], respectively (Figure 4). Bootstrap resampling keeps the selected positions of each prompt together. Because the temperature study uses a smaller state set, its $T = 1$ estimate refers to a different collection of states from the 5,504-state default study.

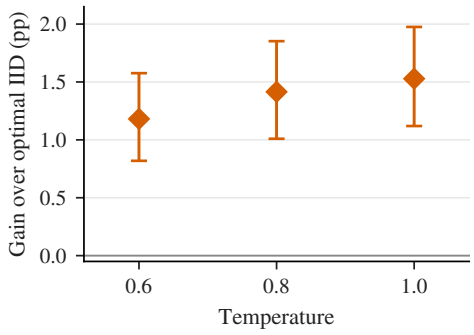

Figure 4: **Stratified improves on optimal IID across the tested temperatures.** Qwen3-4B/0.6B, $n = 4$, $K = 100$, 384 common states per temperature. Points are macro gains in percentage points; bars are paired prompt-cluster 95% intervals. The target and draft temperatures change together.

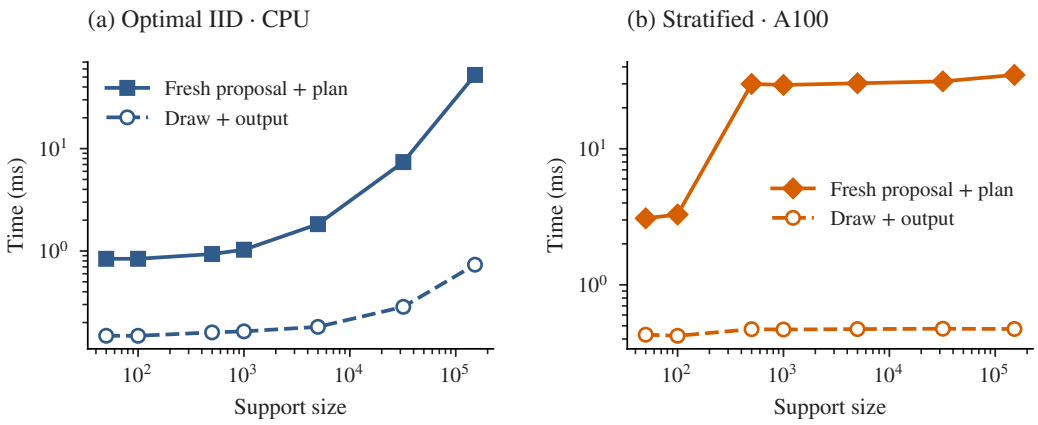

Figure 5: **Fresh construction and output sampling remain tractable at vocabulary scale.** Synthetic-support scaling at $n = 4$, medians of three repetitions, logarithmic axes. Solid lines include proposal preparation and the full target-dependent plan; dashed lines include proposal drawing and output sampling. CPU and A100 measurements are separate implementation profiles.

E.6 Construction and online scaling

Figure 5 reports construction and draw/output times at $n = 4$ over support sizes from 50 to 151,936. We generate heavy-tailed synthetic inputs using a fixed random seed (190932): $q_i \propto i^{-1.15}$ and $p_i \propto q_i u_i$, with independent $u_i \sim \mathrm{Uniform}(0.25, 2)$. Each setting has three repetitions. The construction measurement includes a fresh proposal distribution and a complete target-dependent plan. Each repetition times 32 independent proposal-draw/output calls. Plot points are medians of all three repetitions, including the first observed call.

Optimal IID is measured on CPU and Stratified on an A100, shown in separate panels. At the largest support, their fresh construction times are 52.45 and 34.89 ms, respectively; their draw/output times are 0.735 and 0.474 ms. The panels characterize each implementation's scaling on its specified device. The language-model decoding measurement below includes a new distribution and plan as decoding advances.

Table 6: **Stratified's verification construction and sampling average less than one millisecond per output token.** Independent GH200 profile; all entries are ms/output-token. The first four rows are disjoint. The verification subtotal and method total summarize those rows and must not be added again. Totals are computed before rounding.

Component	Sequential IID	Stratified
Draft support and proposal preparation	0.370	0.547
Target support and tail mass	0.181	0.186
Verification-plan preparation	0.068	0.181
Online construction, sampling, and correction	0.432	0.308
Verification construction + sampling	0.501	0.489
All method components	1.052	1.222

E.7 SINGLE-GH200 DECODING PROTOCOL

The latency comparison uses Qwen3-32B as target and Qwen3-1.7B as draft, with one GH200 per trajectory, tensor parallelism one, batch size one, BF16 execution, temperature 1, full-vocabulary target sampling, and draft support $K = 100$. We compare AR, Sequential IID with four candidates, and Stratified with four depth-one candidates. Residual correction retains the full-vocabulary target mass.

Stratified orders tokens by decreasing draft probability. This order depends on q, not on the current target distribution, so the same-marginal coverage guarantee still applies.

Each method generates 128 tokens on 164 HumanEval, 100 GSM8K, and 80 MT-Bench prompts with seeds zero and one, giving 688 trajectories per method and 2,064 in total. All 344 prompts enter the reported results. The timer includes proposal, verification-plan construction, online sampling and full residual correction, model execution, KV and draft-state recovery, synchronization, and control overhead. Model loading, compilation, warmup, graph capture, and prompt prefill are outside this steady-state decode measurement. These runs do not include stage-level profiling.

Latency is averaged within each dataset, then equally across datasets. Throughput ratios divide the baseline's macro latency by Stratified's macro latency. Paired uncertainty uses 10,000 within-dataset bootstrap resamples of prompts, preserving both seeds and all methods in each prompt cluster. The 95% intervals are [1.5758, 1.5892] relative to AR and [1.0063, 1.0122] relative to Sequential IID. The per-dataset ratios relative to Sequential IID are 1.0044 on GSM8K, 1.0112 on HumanEval, and 1.0118 on MT-Bench.

Pooling decoding rounds, observed membership rises from 88.33% for Sequential IID to 91.32% for Stratified, and output tokens per round rise from 1.876 to 1.908. Output per round increases by 1.68%, while complete wall time per round increases by 0.74%. These pooled quantities use different weights from the macro latency ratio. The observed memberships are measured on each method's own trajectories; the acceptance frontiers instead compare probability laws at common states.

E.8 VERIFICATION CONSTRUCTION AND ONLINE SAMPLING COST

Table 6 uses an independent profile of 12 prompts, two seeds, and both multi-draft methods (48 trajectories). We measure disjoint wall-clock intervals on the host, synchronizing at stage boundaries after preceding model work. The verification total includes all construction work across the preparation and online stages, candidate selection, full residual correction, and output readout. Target-model forward execution is outside these method components.

The profile produces 3,072 Stratified output tokens in 1,625 verification rounds. Thus 0.489 ms/output-token, including bonus tokens in the denominator, corresponds to 0.925 ms/verification round. Proposal and target-support preparation raise the full method cost to 1.222 ms/output-token. Instrumentation increases Stratified's latency on these inputs from 13.472 to 14.274 ms/output-token (5.95%); the main end-to-end result uses the separate uninstrumented runs. The profiled and uninstrumented runs produce the same outputs for matched seeds.

Table 7: **The production token order retains the gain beyond optimal IID.** Expected membership (%), equally weighted across datasets, on 1,376 matched states from Stratified trajectories.

Method	Expected membership
Sequential IID	84.41
Optimal IID	85.82
Stratified (canonical order)	86.53
Stratified (production order)	86.54
Marginal Oracle	86.60

Reported Global Resolution costs. Global Resolution reports 38.44 ms/token and a 14% solver success rate on Llama-3 at $K = 100$, $n = 3$, and tolerance 10^{-4}; early termination triggers fallback in its budgeted evaluation (Thomas & Pal, 2026, Table 1 and Section 7.2). Its CPU optimization timings and our construction-plus-sampling profile differ in hardware, models, and token amortization. We use these reported costs as context, not as a matched speedup measurement.

E.9 Connecting the production proposal to the acceptance frontier

We evaluate both increasing-token-ID (canonical) and decreasing-q (production) orders on 1,376 common states: positions $0, 32, 64, 96$ of Stratified's seed-zero trajectories for all 344 prompts. This is a separate state collection from the AR-prefix frontier in Table 1. Each method receives the same full target and draft logits, with $n = 4$, $K = 100$, and temperature 1.

Stratified with the production token order exceeds optimal IID by 0.72 pp (95% CI [0.58, 0.87]) and captures 92.9% of the remaining marginal headroom (95% CI [89.1, 96.2]). The canonical order captures 91.5%; changing to production order adds 0.011 pp of membership. These intervals retain each prompt's four positions in the same bootstrap cluster. The comparison establishes acceptance, not end-to-end latency for Optimal IID or the Marginal Oracle.

Artifact reproduction. The accompanying source package includes numeric summaries and scripts to reproduce the tables and figures. The code supplement provides the core implementations, experiment drivers, configurations, and prompt manifests.