DOES A GRAMMAR-CONSTRAINED SPECULATIVE VERIFIER SAMPLE THE CONDITIONAL LAW? FIDELITY BOUNDS, NUMERICAL PATHS AND EXACT ACQUISITION

Anonymous authors
Paper under double-blind review

ABSTRACT

A speculative verifier samples whatever target it is supplied, so grammar-constrained speculative decoding samples the model's grammar-conditional law only if the verifier receives the future-validity-corrected target of grammar-aligned decoding. We measure what an implemented stochastic verifier actually samples once that target is supplied, on finite canonical JSON events whose conditional law is enumerated exactly. In a Qwen3-8B/DFlash block verifier, simultaneous 95% bounds place the terminal law within 0.028 total variation (TV) of the conditional law on four events, removing at least 88% of each masking gap; a registered chain-draft verifier and a second target, Qwen3-14B, reach the same conclusion, and a crossed dtype $\times$ execution-path family places every rejection in the bf16 native block forward. We show that the masking gap is the relative dispersion of the legal mass along sampled paths and cannot exceed one minus the event's probability. On 25 matched events, stating the schema in the prompt lowers the median gap from 0.28–0.48 to 0.009–0.051, and across instances the gap follows the visit-weighted dispersion of future validity. The correction itself costs the verifier 0.03 per-position acceptance. To acquire the target exactly, we prove that bounds from acquired rows alone certify a resolved-leaf sampler to within a factor two of the best terminal guarantee those rows allow once valid mass is resolved, and that draft span proposals sample the conditional law for every reachable cache state; adaptive exact samplers need as little as 0.019 of full enumeration's wall clock, and span proposals save target forwards when each forward recomputes a long prompt.

1 INTRODUCTION

Structured-output systems combine local vocabulary masking with speculative decoding to generate JSON, SQL or code more quickly (Dong et al., 2025; Li et al., 2026; Willard & Louf, 2023; Leviathan et al., 2023; Chen et al., 2023). A valid output and a sample from the model conditioned on validity are different requirements. Local masking renormalizes at each prefix; a speculative accept/reject step preserves that supplied distribution. Assessing conditional-law fidelity therefore requires specifying the terminal event, the target passed to the verifier and the numerical path that evaluates it.

Grammar-Aligned Decoding (GAD) (Park et al., 2024) established the distinction between the locally projected law μ^{proj} and the globally conditional law $\mu^{\star}$. Its correction weights each candidate by expected future grammaticality, denoted here by $\Phi_t(y) = \Pr_{z\sim p}[x_{<t}yz \in \mathcal{L}(\mathcal{C})]$. The corrected kernel is $\mu_t^{\star}(y) \propto p_t(y)\Phi_t(y)$. We ask an operational question: when this correction is supplied to a real block verifier, do the implemented samples match the declared conditional, and which resources are required to obtain the correction? We answer on finite languages whose laws can be enumerated, with crossed precision and execution-path controls, and we state every result for the event on which it is measured: a canonical leaf-stopping token trie is a narrower event than unrestricted valid JSON.

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.

What is new. Three lines of prior work meet here without answering this question. GAD and its ASAp sampler (Park et al., 2024) introduce Φ_t and evaluate their samplers by KL to a conditional estimated from samples; Alpay & Senturk (2026) give the Doob form and its exactness condition. CARS (Parys et al., 2026) samples $\mu^\star$ exactly, but without a draft or a verifier. The supplied-target identity (Leviathan et al., 2023, App. A.1) states what an ideal verifier samples, not what an implemented one does in finite precision. None of them measures the terminal law of an implemented stochastic verifier against an enumerated $\mu^\star$ with a stated finite-sample resolution, or separates estimator error, floating-point path and acquisition cost; this paper does, and Section 2 contrasts it with speculative steps inside constrained generation (Melcer et al., 2025; Nakshatri et al., 2025).

Contributions.

1. *Fidelity with a stated resolution, and where it is lost.* Supplying the exact corrected target to a Qwen3-8B/DFlash-b16 verifier lowers mean terminal TV on four canonical JSON tries from 0.355 to 0.0072 in fp32; with 95% simultaneous confidence its law is within 0.013–0.028 of $\mu^\star$, removing at least 88% of masking gaps of 0.19–0.62. Registered extensions give a chain-draft verifier within 0.019–0.041 and a second target, Qwen3-14B, within 0.036–0.050 on cross-field events with gaps 0.30–0.85, and six schema-stating prompt cells within 0.027–0.084 against gaps of 0.11–0.41. A crossed dtype $\times$ execution-path family places all four Holm rejections in bf16 native block-forward cells, while fp32 block-forward and reference paths return identical counts in Section 5.1.
2. *When the correction matters, and what it costs the verifier.* The masking gap is the relative dispersion of the legal mass along paths and is at most $1 - \Pr_p(\mathcal{C})$ by Proposition 5. On 25 events under matched prompts, stating the schema lowers the median gap from 0.28–0.48 to 0.009–0.051; across the 150 instances the gap tracks the visit-weighted dispersion of future validity with Spearman correlation 0.99. In the verifier of Section 3, the correction changes per-position acceptance by at most the local masking distortion, and by 0.032 on average in the fp32 runs.
3. *Exact acquisition of the corrected target.* Acquired rows give a local midpoint TV bound in Proposition 8, with no violation in 30,576 comparisons. Separately, the resolved-leaf terminal certificate $R/(L+R)$ is within a factor two of the best guarantee from those rows by Proposition 9, and is never violated along 180 query traces. Adaptive exact samplers need 0.019–0.655 of full enumeration's wall clock at supports 64 and 256; draft span proposals (Spec-4), exact for every reachable cache state by Proposition 11 and by exhaustive trajectory enumeration, batch a span's rows into one target forward and are faster than CARS in all 6 resolved long-prompt cells of Section 5.2, where prompt recomputation dominates.

2 Related Work

What a speculative verifier preserves. Draft-then-verify decoding (Leviathan et al., 2023; Chen et al., 2023) uses tree (Miao et al., 2024), feature-level (Li et al., 2024), multi-head (Cai et al., 2024) and block-diffusion (Chen et al., 2026) drafters; optimal-transport (Sun et al., 2023) and joint block verification (Sun et al., 2025) change the coupling. For the sequential per-token rule, Appendix A.1 of Leviathan et al. (2023) shows that the committed token has the law of whatever target the verifier is given; Corollary 2 and Proposition 4 apply this identity to a masked and to a corrected target.

The grammar-conditional law and future validity. Grammar-Aligned Decoding (GAD) (Park et al., 2024) showed that local masking samples a law different from the model conditioned on grammaticality, and introduced the expected-future-grammaticality statistic that we write as Φ_t; its ASAp sampler refines that statistic across repeated autoregressive samples, and is evaluated by KL to an estimate of the conditional built from the observed samples. AWRS (Lipkin et al., 2025) states the same global bias and supplies unbiased local weights for sequential Monte Carlo (SMC). Alpay & Senturk (2026) characterize the conditional kernel as a Doob h-transform with survival probability h in their Theorem 5, prove in Theorem 6 that masking equals conditioning iff h is constant over admissible tokens, and bound one-step KL and TV distortion by the survival-probability spread in Proposition 4 and Corollary 2. We measure, beyond these analyses, the terminal law of an implemented stochastic verifier given the corrected target, and its dependence on estimator error, floating-point path and prompt interface.

Samplers of the constrained conditional. Exact or asymptotically exact routes to $\mu^\star$ include ASAp, twisted SMC (Zhao et al., 2024) and its grammar instantiation (Loula et al., 2025), dynamic importance sampling (Ye et al., 2025), MCMC (Anaya Gonzalez et al., 2025), amortized GFlowNet samplers (Hu et al., 2024), and CARS (Parys et al., 2026), which samples $\mu^\star$ exactly while pruning every invalid continuation it meets. Recent and concurrent work reuses CARS certificates across multi-step histories (Shihab et al., 2026), builds tractable proposals that carry future constrained mass (Dang et al., 2026), learns HMM surrogates whose cached backward messages answer continuation queries (Yidou-Weng et al., 2025), approximately corrects logits toward $\mu^\star$ from parser state (Özgü et al., 2026), or prunes with precomputed pushdown reachability distances (Collura et al., 2026); the last targets validity and task quality rather than the conditional law. Best-first playouts refine GAD's approximation (Susini & Cazenave, 2026) without sampling $\mu^\star$ exactly. Exact Φ_t is a counting quantity. Prefix probabilities of probabilistic CFGs follow from inside-style recursions (Stolcke, 1995; Goodman, 1999), but a transformer does not factorize over grammar states; counting the length-n words of a nondeterministic finite automaton is already #P-hard (Álvarez & Jenner, 1993). Under the uniform distribution, approximation is easier: after a quasi-polynomial sampler (Gore et al., 1997), Meel & de Colnet (2026) give an FPRAS for counting the length-n words of a context-free grammar and a polynomial-time almost-uniform sampler. These results concern uniform weights, whereas Φ is continuation mass under an autoregressive model, and computing it exactly is #P-hard by Proposition 7. CARS never forms Φ; Section 5.2 compares these exact routes in wall clock.

Speculation and lookahead inside constrained generation. AprAD (Melcer et al., 2025) reuses the accept/residual step between the error-adjusted model before and after a newly found violation, with no separate draft and an approximate target law. CDSL (Nakshatri et al., 2025) verifies greedy draft lookaheads with target and reward thresholds, reporting speed-up and constraint satisfaction. DOMINO (Beurer-Kellner et al., 2024) combines subword-aligned masking with speculative execution; future-aware reweighting also appears in controlled generation (Yang & Klein, 2021; Lu et al., 2022; Mudgal et al., 2024), without an exact survival table or a verifier.

Numerical losslessness. Chen et al. (2023) qualify the preservation of the target law as holding up to hardware numerics, and Yuan et al. (2025) show that precision, batch size and GPU configuration change sampled and greedy outputs of reasoning models. Koziev et al. (2026) report that Orthrus reproduces the greedy reference trajectory on all prompts in FP32 but on 43–45% under BF16, with unchanged benchmark accuracy. Greedy agreement does not test a stochastic law; our crossed dtype $\times$ execution-path family tests the terminal law of a stochastic verifier and localizes the rejected cells to the native bf16 block forward in Section 5.1.

3 What the Verifier Samples, and What It Should Sample

Setup. Let V be a finite vocabulary and p a target language model with $p_t(y) = p(y \mid x_{<t})$. Fix a bounded stopping rule $\tau \le T$ and an acceptance event $\mathcal{C}$ on the stopped token path. Stopping can be EOS/horizon termination or arrival at an accepted leaf of a prefix-free finite token trie. In the latter case $p(x) = \prod_{t \le |x|} p_t(x_t)$ is prefix-cylinder mass: the sampler stops at the leaf and includes no EOS factor. These are different conditioning events. Our JSON experiments use only one canonical tokenization per string, out of 1,040 to at least 10^7 under the Qwen3 tokenizer as counted in Section A, not all tokenizations or all JSON whitespace variants. Dyck experiments include their specified EOS/horizon rule. Let A_t be the constraint's admissible next tokens, $Z_p = \sum_{y \in A_t} p_t(y) > 0$, and $Z_{\mathcal{C}} = \Pr_p(\mathcal{C}) > 0$. Complete paths are padded by an absorbing terminal state in products to T.

Definition 1 (Locally projected and grammar-conditional laws)**.**

$$\mu_t^{proj}(y) = p_t(y)\mathbf{1}[y \in A_t]/Z_p,$$
$$\mu^{proj}(x_{1:T}) = \textstyle\prod_t \mu_t^{proj}(x_t \mid x_{<t}),$$
$$\mu^\star(x_{1:T}) = p(x_{1:T})\mathbf{1}[x_{1:T} \in \mathcal{L}(\mathcal{C})]/Z_{\mathcal{C}}.$$

Future validity. The correction statistic is GAD's expected future grammaticality written per candidate token,

$$\Phi_t(y \mid x_{<t}) = \Pr_{z \sim p}\big[x_{<t}\, y\, z_{t+1:T} \in \mathcal{L}(\mathcal{C})\big],$$

and the grammar-conditional kernel is the Doob h-transform (Doob, 1984; Park et al., 2024; Alpay & Senturk, 2026)

$$\mu_t^\star(y \mid x_{<t}) = \frac{p_t(y)\,\Phi_t(y \mid x_{<t})}{\sum_{y' \in A_t} p_t(y')\,\Phi_t(y' \mid x_{<t})},$$
$$\bar{\Phi}_t := \mathbb{E}_{\mu_t^{\text{proj}}}[\Phi_t] = \frac{\sum_{y \in A_t} p_t(y)\Phi_t(y)}{Z_p}.$$

3.1 The verifier preserves its target

Consider a verifier that receives a draft token d from a proposal $\tilde{q}_t$ and applies the standard per-token rule: accept d with probability $\min\{1, \tilde{p}_t(d)/\tilde{q}_t(d)\}$, otherwise sample from the residual $(\tilde{p}_t - \tilde{q}_t)^+$ renormalized (Leviathan et al., 2023; Chen et al., 2023). A grammar-constrained decoder is in the class **LMS** if it meets three conditions: B1, it accesses the constraint only through A_t; B2, it applies this rule with $\tilde{p}_t = \mu_t^{\text{proj}}$; and B3, it maintains the exact constraint state under accept/rollback.

Corollary 2 (Preservation of an explicitly masked target)**.** *Every LMS decoder has per-step kernel μ_t^{proj} on every committed prefix, hence samples μ^{proj}; if $\mu^{proj} \neq \mu^\star$ it does not sample $\mu^\star$.*

This is the supplied-target identity in Appendix A.1 of Leviathan et al. (2023) applied to the masked target plus the chain rule, as proved in Section F.1; it concerns the supplied target, and B1 does not imply B2, so samplers that supply a different target are not covered. Its scope is the sequential per-token rule: chain drafters satisfy it directly, and a block drafter does when each position is verified against its actual conditional proposal given the drafted prefix, as in our DFlash implementation of Section 5.1; tree verification is not covered. Because verification is lossless for its supplied target, adding a mask does not make the target $\mu^\star$. We call the verifier arm that supplies $\tilde{p}_t = \mu_t^{\text{proj}}$ *SchemaTV* and the arm that supplies $\tilde{p}_t = \mu_t^\star$ *exact-*Φ.

3.2 Correcting the target

Proposition 3 (Exactness condition, after Alpay & Senturk (2026))**.** *$\mu_t^{proj} = \mu_t^\star$ iff Φ_t is constant on $\{y \in A_t : p_t(y) > 0\}$.*

Proof. On that set $\mu_t^\star(y)/\mu_t^{\text{proj}}(y) = \Phi_t(y)Z_p/Z^\star$ with $Z^\star = \sum_{y \in A_t} p_t(y)\Phi_t(y)$, which is identically one iff Φ_t is constant there. Alpay & Senturk (2026) state this over all admissible tokens. □

Proposition 4 (Oracle correction)**.** *A verifier that applies the standard rule with $\tilde{p}_t = \mu_t^\star$ samples $\mu_t^\star$ at every step, hence $\mu^\star$.*

The proof again uses the supplied-target identity in Appendix A.1 of Leviathan et al. (2023). By Section F.5, the per-step gap has the closed form $D_{\mathrm{KL}}(\mu_t^\star \| \mu_t^{\text{proj}}) = \mathbb{E}_{Y \sim \mu_t^\star}[\log(\Phi_t(Y)/\bar{\Phi}_t)]$, the $\mu^\star$-average log-ratio of a token's future validity to the projection-average future validity; averaged along $\mu^\star$-paths, these local terms give the terminal KL.

Proposition 5 (Masking gap as path-mass dispersion)**.** *On a constraint without dead ends, such as a finite trie, let $W(x) = \prod_{t \le |x|} Z_p(x_{<t})$ be the legal mass collected along a path and $Z_\mathcal{C} = \Pr_p(\mathcal{C})$. Then $\mu^\star(x) = \mu^{proj}(x)W(x)/Z_\mathcal{C}$, $\mathbb{E}_{\mu^{proj}}[W] = Z_\mathcal{C}$,*

$$\mathrm{TV}(\mu^{proj}, \mu^\star) = \tfrac{1}{2}\,\mathbb{E}_{\mu^{proj}}\big|W/Z_\mathcal{C} - 1\big| \;\le\; 1 - Z_\mathcal{C},$$
$$D_{\mathrm{KL}}(\mu^\star \| \mu^{proj}) = \textstyle\sum_t \mathbb{E}_{\mu^\star}\big[D_{\mathrm{KL}}(\mu_t^\star \| \mu_t^{proj})\big].$$

The proof is in Section F.5. Equivalently, $Z_\mathcal{C} \le 1 - \mathrm{TV}(\mu^{\text{proj}}, \mu^\star)$: a gap approaching one forces $Z_\mathcal{C}$ towards zero. Rarity alone is insufficient; legal mass must vary across paths, as measured by $D = D_{\mathrm{KL}}(\mu^\star \| \mu^{\text{proj}})$ in Section 5.3.

What is specific to speculative verification. For proposal $\tilde{q}_t$ and supplied target $\tilde{p}_t$ the per-position acceptance probability is $\alpha(\tilde{q}_t, \tilde{p}_t) = \sum_y \min\{\tilde{q}_t(y), \tilde{p}_t(y)\} = 1 - \mathrm{TV}(\tilde{q}_t, \tilde{p}_t)$ (Leviathan et al., 2023), so by the triangle inequality $|\alpha(\tilde{q}_t, \mu_t^\star) - \alpha(\tilde{q}_t, \mu_t^{\text{proj}})| \le \mathrm{TV}(\mu_t^\star, \mu_t^{\text{proj}})$ for every proposal at

the same prefix. Fidelity is set by the supplied target alone; the draft and the correction determine only how often the verifier accepts. In the fp32 DFlash runs the correction lowers mean per-position acceptance on A–D from 0.887 to 0.855 in Table 2, against masking gaps of 0.19–0.62. Two further effects are specific to speculative verification and are measured below: Section 5.1 shows that the block forward is a numerical path distinct from the reference rows, and Section 5.2 that draft spans reduce target forwards when the target is acquired exactly.

3.3 Fidelity certificates for approximate Φ

At prefixes with positive true survival mass, let $\hat{\Phi}_t \geq 0$ be an estimate with $\sum_{y \in A_t} p_t(y)\hat{\Phi}_t(y) > 0$ and $\hat{\mu}_t(y) \propto p_t(y)\hat{\Phi}_t(y)$ on A_t.

Theorem 6 (Per-prefix certificates)**.** *(a) Additive: with* $\bar{\delta}_t := \mathbb{E}_{\mu_t^{proj}}|\hat{\Phi}_t - \Phi_t| \leq \delta_t := \max_{y \in A_t} |\hat{\Phi}_t(y) - \Phi_t(y)|$, $\mathrm{TV}(\hat{\mu}_t, \mu_t^\star) \leq \bar{\delta}_t/\bar{\Phi}_t \leq \delta_t/\bar{\Phi}_t$. *(b) Relative: if* $\hat{\Phi}_t(y) = r_y \Phi_t(y)$ *with* $r_y \geq 0$*, then* $\mathrm{TV}(\hat{\mu}_t, \mu_t^\star) \leq \mathbb{E}_{\mu_t^\star}|r_Y - 1| \leq \max_y |r_y - 1|$. *(c) Sequence: for the induced laws over complete sequences,*

$$\mathrm{TV}(\hat{\mu}, \mu^\star) \leq \sum_t \mathbb{E}_{x_{<t} \sim \hat{\mu}}\Big[\min\Big\{1, \frac{\bar{\delta}_t(x_{<t})}{\bar{\Phi}_t(x_{<t})}\Big\}\Big],$$

and likewise with the relative bound. At a generated prefix with zero true survival, define an arbitrary reference continuation and charge one instead.

The proofs in Section F.2 use the elementary inequality $\mathrm{TV}(a/\sum a, b/\sum b) \leq \sum |a - b| / \sum a$ for non-negative vectors. The forms are informative when $\bar{\delta}_t < \bar{\Phi}_t$; Section C.4 evaluates the full sequence expression on enumerable grammars. Part (c) is what a terminal-law comparison must use; a single root-prefix value is not a certificate for the sequence law.

Proposition 7 (Hardness and exact cost)**.** *Computing* $\Phi_t(y)$ *exactly is #P-hard under polynomial-time Turing reductions for general context-free grammars, already for right-linear ones and a uniform base model; Section F.3 gives the proof. For a finite language represented as a token trie, exact* Φ *for one model, prompt and grammar needs one target probability row per non-terminal trie prefix, i.e.* $O(\#\textit{trie nodes})$ *rows; the table depends on the model and the prompt, not only on the grammar.*

4 A Hierarchy of Φ_t Estimators

Because exact Φ_t is #P-hard in general by Proposition 7, a verifier that needs the full per-candidate correction must approximate it; exact samplers such as CARS avoid the full table by acquiring rows adaptively, as in Section 5.2. We evaluate four tiers against exact $\mu^\star$; every estimator must return a *per-candidate* vector, since a constant $\hat{\Phi}_t \equiv c$ is a logit shift and reproduces μ_t^{proj}.

Tier 0, Uniform. $\hat{\Phi}_t \equiv 1$ gives μ^{proj} at no cost, like XGrammar, Outlines and any LMS decoder.

Tier 1, OneStep. Let $A_{t+1}(x_{<t}y)$ be the valid set after appending y. **OneStep-True** uses the child conditional,

$$\hat{\Phi}_t^{\text{1s-true}}(y) = \sum_{u \in A_{t+1}(x_{<t}y)} p(u \mid x_{<t}y),$$

at one extra target forward per candidate. **OneStep-Cheap** reuses the current row, $\hat{\Phi}_t^{\text{1s-cheap}}(y) = \sum_{u \in A_{t+1}(x_{<t}y)} p_t(u)$, which assumes context independence; it appears only in the Dyck study of Table 23, not in the finite-JSON experiments. OneStep-True is a one-step truncation with the exact identity, proved in Section F.4,

$$\hat{\Phi}_t^{\text{1s-true}}(y) - \Phi_t(y) = \sum_{u \in A_{t+1}(x_{<t}y)} p(u \mid x_{<t}y)\big(1 - \Phi_{t+1}(u \mid x_{<t}y)\big) \geq 0, \tag{1}$$

the probability-weighted future invalidity of valid successors. The error depends on the model, the prompt and the grammar jointly rather than on a grammar class: the toy regular language a^+b^+ has gaps 0.15–0.27 and the budget DFA of Section C has gaps 0.67–0.96.

Tier 2, Monte-Carlo rollout. With k unconstrained rollouts from $x_{<t}y$, $\hat{\Phi}_t^{\mathrm{mc}}(y) = \frac{1}{k}\sum_j \mathbf{1}[\mathrm{rollout}_j \in \mathcal{L}(\mathcal{C})]$, where a rollout counts as valid only if it stays inside the constraint and terminates inside $\mathcal{L}(\mathcal{C})$. When no candidate has a valid rollout the reweighted kernel is undefined, so the fallback is part of the estimator: local projection in the Dyck study, the uniform law on A_t in the finite-JSON sampler of Section C.4. By Hoeffding and a union bound over A_t, with probability $\geq 1-\alpha$, $\delta_t \leq \sqrt{\ln(2|A_t|/\alpha)/(2k)}$, and Theorem 6(a) turns this into a TV bound. Cost is $O(k \cdot h \cdot |A_t|)$ forwards per position for horizon h; this is a reference, not an online decoder.

Tier 3, Exact. Backward dynamic programming over the enumerated prefix graph covers bounded Dyck, finite tries, and regular languages only when the base model factorizes over automaton states, which a transformer does not; its table is per model, prompt and grammar, with cost in Section 5.1.

4.1 An observable stopping rule for a finite query budget

The error in Theorem 6 contains the unknown Φ. A decoder needs a stopping rule computed from the information it has actually acquired. For a finite prefix tree, assign every accepted leaf the interval $[1,1]$, every unresolved nonterminal $[0,1]$, and every known dead end $[0,0]$. Querying a prefix reveals its complete legal-token probability vector; backward Bellman sums then propagate lower and upper survival bounds $L(s), U(s)$. No unqueried candidate is assigned a fictitious observation.

Proposition 8 (Observable candidate-wise certificate)**.** *Fix a viable prefix, one with $\sum_y p_y \Phi_y > 0$, a materialized target vector p, and bounds $0 \leq L_y \leq \Phi_y \leq U_y$ for every legal child. For midpoint weights $m_y = (L_y + U_y)/2$ with $\sum_y p_y m_y > 0$, let $q_y = p_y m_y / \sum_j p_j m_j$. Then*

$$\mathrm{TV}(q, \mu_t^\star) \leq b_t := \frac{\sum_y p_y (U_y - L_y)}{\sum_y p_y (U_y + L_y)} \leq 1.$$

The bounds remain valid after any adaptive sequence of exact prefix queries. Summing b_t along a fixed decoder's generated-prefix distribution bounds its terminal law.

The proof is in Appendix F.6. The ratio needs no exact-Φ labels and is nonvacuous once some positive lower mass is resolved. Two samplers use these bounds. The *midpoint kernel* q carries the per-step certificate b_t; on tries A and B its 30,576 local comparisons with the exact law show no violation in Section C.5. The *resolved-leaf sampler* q_{leaf} draws only among resolved leaves and carries the terminal certificate $R/(L+R)$, with L the resolved valid-leaf mass and R the prefix mass of the unqueried frontier, as shown in Section F.6.

Proposition 9 (Information bound)**.** *Let $L > 0$, let every non-terminal prefix have a next token outside the event, as for every event studied here, and let m be the fewest valid leaves below an unqueried frontier prefix. Every sampler q computed from the acquired rows and the topology has $\sup \mathrm{TV}(q, \mu^\star) \geq \frac{m}{m+1}\frac{R}{L+R}$, the supremum over all next-token distributions at the unqueried prefixes, and q_{leaf} attains $R/(L+R)$.*

No certificate from the same rows can thus be below half of $R/(L+R)$; Section F.6 proves this and checks it on 300 small tries. It concerns the resolved-leaf terminal certificate from given rows; it does not bound the midpoint certificate b_t, its sum, the query policy or exact sampling. The certificate is never violated at any step of 180 registered query traces on 20 unseen schemas in Table 26 or of 150 panel traces in Section 5.3, where at first certification of 0.02 the actual TV has median 0.003–0.017 with the schema stated, as Table 9 lists.

Numerical target and estimator error are separate. If the kernel K_t realized by the numerical verifier satisfies $\mathrm{TV}(K_t, q_t) \leq e_t$ at each reachable prefix, then $\mathrm{TV}(K, \mu^\star) \leq \min\{1, \sum_t \mathbb{E}_K[\min\{1, e_t + b_t\}]\}$, so an exact future table, for which $b_t = 0$, does not certify a verifier whose block-forward probabilities differ from the reference rows.

5 Experiments

All fidelity numbers are distances to a law that is enumerated exactly; sampled runs report plug-in TV from N terminal samples. Table 1 lists what each main result compares; the estimator hierarchy with exact induced laws of Section C.4 and further experiments are in Section C.

Table 1: Evidence map: event, reference law, sampler, statistical unit and cost measure.

Result	Table	Event	Reference / sampler	Unit	Cost measure
Correction, bounds	3, 2	tries A–D	exact $\mu^\star$ / DFlash, fp32	4 seeds $\times 10^4$	table rows
Second verifier	15	tries A–D	exact $\mu^\star$ / chain draft, fp32	4 cells $\times 10^4$	—
Second target	16	events F–I	exact $\mu^\star$ / chain draft	8 cells $\times 8{\cdot}10^3$	—
Numerical path	14, 12	tries A–E	same-dtype $\mu^\star$ / 2 paths	40 cells, one Holm test	—
Gap size, prompts	6, 7	60 + 150 inst.	exact $\mu^\star$, μ^{proj}	20 / 25 event clusters	—
Stopping bound	26	60 instances	exact $\mu^\star$ / acquired rows	180 traces	target rows
Row-cache cost	4	8 events	all arms exact, fp32	2 seeds $\times$ 7 timings	wall clock

Table 2: The headline cells on the fp32 path, with a Qwen3-8B target, a DFlash-b16 draft and $N = 10{,}000$ per arm. Gap is the analytic $\mathrm{TV}(\mu^{\text{proj}}, \mu^\star)$ of the run's own reference; SchemaTV and exact-Φ are the sampled arms' TV to $\mu^\star$, with the 99% multinomial quantile of the arm's own null in parentheses; acceptance is per position. * marks a Holm-adjusted rejection in the 40-cell family; bf16 columns: same-session native cells of Table 14; last column: TV between the fp32 and bf16 reference laws.

Schema	Str.	fp32 native path				bf16 native path, same session		$\mathrm{TV}(\mu^\star_{32}, \mu^\star_{16})$
		Gap	SchemaTV	exact-Φ (99% q.)	Accept SchemaTV/Φ	SchemaTV	exact-Φ (99% q.)	
A	3	0.193	0.192	0.0062 (0.0083)	0.824/0.775	0.189*	0.0106 (0.0080)*	0.011
B	4	0.428	0.427	0.0012 (0.0125)	0.970/0.937	0.437	0.0379 (0.0129)*	0.042
C	18	0.187	0.186	0.0082 (0.0163)	0.879/0.863	0.199*	0.0166 (0.0160)	0.030
D	24	0.616	0.616	0.0134 (0.0165)	0.876/0.843	0.625	0.0102 (0.0165)	0.019
Mean A–D	—	0.356	0.355	0.0072	0.887/0.855	0.363	0.0188	—
E	2,000	0.231	0.230	0.0269 (0.0377)	0.834/0.832	0.220	0.0469 (0.0369)	0.017

5.1 Exact correction inside a DFlash verifier

Implementation. The target is Qwen3-8B and the draft is z-lab/Qwen3-8B-DFlash-b16. The draft proposes a block from per-position marginals computed in one parallel pass, each masked to the drafted prefix's trie children and renormalized; the target scores the block in one forward; positions are verified left to right with the rule of Corollary 2 against the masked target for SchemaTV or the Φ-reweighted masked target for exact-Φ, stopping at the first rejection with a residual sample, and taking a bonus token from the target when the whole block is accepted. Exact Φ is the backward-DP table of Proposition 7, rebuilt for every prompt at one target forward per non-terminal prefix, 7, 16, 70, 104 and 2,232 for the tries A–E of Section A with 3, 4, 18, 24 and 2,000 strings.

Correction removes the masking gap. On the fp32 path the exact-Φ target lowers mean terminal TV on A–D from 0.355 to 0.0072, with every corrected cell inside its pointwise 99% multinomial envelope in Table 2. Three further pre-registered fp32 seeds give mean corrected TV 0.0053, 0.0085 and 0.0078 on A–D and no Holm-rejected corrected cell, and an independent rerun reproduces all 20 fp32 count vectors of the five tries in Table 12.

Finite-sample fidelity bounds. A non-rejection is not an equivalence statement, so we bound the terminal law K of the fp32 corrected verifier directly. The four seeds sample the same kernel, and the pooled empirical law has TV to $\mu^\star$ at most the mean of the four per-seed TVs. For n draws over k atoms, Hoeffding over events gives $\Pr[\mathrm{TV}(\hat{p}, p) \geq \varepsilon] \leq (2^k - 2)e^{-2n\varepsilon^2}$ (Weissman et al., 2003). With a Bonferroni split of $\delta = 0.05$ over the five tries, Table 3 gives, simultaneously with 95% confidence and rounded up, $\mathrm{TV}(K, \mu^\star) \leq 0.013$, 0.016, 0.023 and 0.028 on A–D, against masking gaps of 0.19–0.62, so the verifier removes at least 88% of each small try's gap. On E the union bound over 2^{2000} events gives 0.163; a post hoc head/tail bound that keeps the h atoms of largest $\mu^\star$ mass and lumps the rest, adding its mass, gives $\mathrm{TV}(K, \mu^\star) \leq 0.059$ in Section A.1, removing at least 74% of E's gap of 0.231. A second post hoc bound puts the verifier's probability of each of E's 1,002 field–value events within 0.015 of $\mu^\star$, against a largest masking error of 0.193 in Table 22.

A second verifier. A second implementation, described in Section B and registered before its first run, replaces the block draft by an autoregressive Qwen3-1.7B chain draft of four tokens per round.

Table 3: Terminal-law bounds for the fp32 exact-Φ DFlash verifier. Masking gap is $\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^\star)$; the bounds hold jointly with probability at least 0.95 by Bonferroni over the five tries. The post hoc field–value column is a simultaneous 95% bound on $\max_e |K(e) - \mu^\star(e)|$ over the try's field–value events, by Hoeffding with Bonferroni over the 1,026 events of A–E. ‡Post hoc head/tail bound of Section A.1 on the full terminal law at the same δ per try; the registered bound on E is 0.163.

Try	Atoms	Masking gap	Corrected TV, seeds 42/1/2/3	N	95% bound	Field–value
A	3	0.193	0.0062/0.0012/0.0056/0.0030	40,000	0.013	0.016
B	4	0.428	0.0012/0.0025/0.0074/0.0122	40,000	0.016	0.015
C	18	0.187	0.0082/0.0082/0.0044/0.0090	40,000	0.023	0.016
D	24	0.616	0.0134/0.0094/0.0166/0.0070	40,000	0.028	0.014
E	2,000	0.231	0.0269/0.0323/0.0334/0.0318	40,000	0.059‡	0.015

On A–D its fp32 corrected arm has TV 0.0013, 0.0030, 0.0068 and 0.0081 to $\mu^\star$ at $N = 10{,}000$ each, with simultaneous 95% bounds 0.019, 0.022, 0.036 and 0.041, removing at least 80% of each masking gap at per-position acceptance 0.86–0.94; in Table 15, neither registered fp32 family has a Holm rejection, and the only bf16 rejection is again in a native cell.

A second target and cross-field events. A registered extension in Table 16 runs this verifier with a Qwen3-14B target in fp32, and Qwen3-8B, on four new events of 10–25 atoms with cross-field constraints: a city within its country, a start before its end, a sum field equal to $a + b$, and a nested object. No cell is Holm-rejected, and every cell whose gap exceeds the resolution of 0.05 is supported: the corrected law is within 0.036–0.050 of $\mu^\star$ for Qwen3-14B and 0.034–0.049 for Qwen3-8B, removing at least 82% of gaps 0.19–0.85. Six schema-stating cells on Qwen3-8B have no rejection and bounds 0.027–0.084 against gaps 0.11–0.41 in Section B.2; on Qwen3-14B they have gaps of 0.014–0.150 and no rejection, and the bound 0.056 removes at least 63% of the largest gap.

Numerical path. The 40-cell family on A–E in Table 14 crosses dtype, native or canonical verifier path and corrected or uncorrected arm, with $N = 10{,}000$ per cell and one Holm family, and localizes the bf16 residual. All four rejections are bf16 native cells, as are all seven cells outside their pointwise 99% envelopes; no bf16 canonical or fp32 cell is rejected, and the fp32 native and canonical paths return identical counts on all five tries. The native path uses the block-forward probabilities and the canonical path the reference rows, so the rejected cells isolate a numerical component in the bf16 block forward; the fp32 forward with fp32 normalization is within 3.0×10^{-6} TV of an fp64 forward on every try in Table 11. As an in-sample description, a registered replay of the logged bf16 native kernels, averaged per node over visits, accounts in Table 13 for at least 80% of the residual TV in three of the four rejected cells and for 67% in the fourth; the localization rests on the fixed references.

5.2 Exact samplers from an empty row cache

We run seven exact samplers of $\mu^\star$ with configurations fixed in advance on eight real-model events, binary-string JSON objects with $|\mathcal{L}| \in \{4, 16, 64, 256\}$, each under a short and a long prompt, Qwen3-8B target and Qwen3-1.7B draft in fp32, for 1 or 16 outputs per history. Each of the 224 histories from two sampling seeds is timed in seven arm-order rotations, which reproduce identical outputs and serve as timing repeats. Table 4 shows CARS (Parys et al., 2026), full enumeration with suffix-shared prefix rows, and Spec-4, the variant of Algorithm 1 that proposes four-token spans from the draft at branching nodes and is exact by Proposition 11 and by exhaustive enumeration in Section E.3; four further arms are recorded. The row backend recomputes the full prefix at every forward without a KV cache, so row cost grows with the prompt; Section D states the timing contract.

Two regularities hold across the matrix. First, adaptive exact sampling pays once the support is moderate: at support 64 and 256 the faster of CARS and Spec-4 needs 0.019–0.655 of the enumeration wall clock, down to 0.019 at support 256 with the long prompt. Second, draft proposals pay when target forwards are expensive: timing repeats agree to within a log-ratio standard deviation of 1.5%, and in all 6 long-prompt cells where both sampling seeds agree, Spec-4 is faster than CARS, with Spec-4/CARS ratios of 0.568–0.814. The saving is in target forwards: Spec-4 scores a drafted span in one forward, using 6–7 target forwards against 16–20 for CARS at support 256 with the long prompt

Table 4: Wall clock of exact samplers on the qualified row backend without a KV cache, from an empty row cache with the models loaded, for 1 and 16 outputs per history. Enumeration seconds are medians; ratios are paired geometric means over 2 seeds × 7 timing rotations; all arms sample $\mu^\star$ exactly. †: the winner's advantage changes sign between the two seeds.

		Enum. wall, s		CARS/Enum.		Spec-4/Enum.		Spec-4/CARS		Fastest
Support	Prompt	1	16	1	16	1	16	1	16	1 / 16 outputs
4	short	0.41	0.41	0.846	1.408	1.119	1.437	1.323	1.021	CARS† / Enum.
4	long	1.66	1.66	0.842	1.544	0.685	1.233	0.814	0.799	Spec-4 / Enum.
16	short	1.52	1.53	0.694	1.066	0.527	1.188	0.760	1.114	Spec-4 / Enum.
16	long	6.53	6.53	0.305	1.314	0.232	0.902	0.760	0.687	Spec-4 / Spec-4
64	short	6.01	6.02	0.125	0.655	0.140	0.659	1.122	1.006	CARS / CARS†
64	long	25.92	25.94	0.107	0.422	0.117	0.367	1.095	0.871	CARS† / Spec-4†
256	short	23.69	23.75	0.040	0.171	0.075	0.271	1.857	1.584	CARS / CARS
256	long	104.01	104.03	0.034	0.067	0.019	0.046	0.568	0.687	Spec-4 / Spec-4

and 16 outputs, at a similar number of rows, 24–29 against 19–23 in Table 27, which pays when each forward recomputes the prompt. A fit to all 1,568 timed runs in Section C.7, with $R^2 = 0.9998$, makes this explicit: prompt tokens are 81–82% of the target cost at long prompts, and with the prompt paid once per history Spec-4 is modelled cheaper than CARS in a long-prompt cell once a target forward's fixed cost exceeds 1.0–5.9 draft forwards, as listed in Table 28.

5.3 Where the gap is large

Unseen schemas. Before any model call we registered 20 rule-generated schemas of four structural families from Section A, three prompts, Qwen3-8B fp32 canonical tables as the reference, and the gap decision rule of Table 30. It is met: the median is 0.465 (95% interval 0.388–0.756) and all 60 schema–prompt instances exceed 0.02, ranging over 0.057–0.999. The prompts do not state the schema, and the conditioning events are rare under them: Φ_{root} ranges from 7×10^{-38} to 8×10^{-9}. A registered crossover changes only the prompt on the 25 events A–E and U01–U20: stating the schema raises the median Φ_{root} of Qwen3-8B from 2×10^{-18} to 0.15–0.31 and lowers its median gap from 0.476 to 0.051 and 0.029, with paired 95% lower bounds 0.17 and 0.19, and that of Qwen3.5-9B from 0.277 to 0.029 and 0.009; in Table 7, 2–4 of 25 events per model and prompt keep gaps ≥ 0.10.

Rarity and dispersion. All 150 panel instances satisfy Proposition 5. Across the panel, chiefly between prompts, the gap increases with rarity: a registered secondary analysis gives Spearman 0.66 with $-\log \Phi_{\mathrm{root}}$, event-bootstrap 95% interval 0.59–0.72. Within every model–prompt group of Table 8, it follows the visit-weighted dispersion D, computed from the same two laws, with Spearman 0.90–0.99. The correction matters where the prompt leaves the event rare and future validity uneven across candidates. The exact cost falls with the gap: CARS draws one exact output from a median 0.15–0.33 of the table's rows with the schema stated and 0.56–0.65 without in Table 9. The gap sits in single fields: the largest field–value error is a median 0.96 of it; with the schema stated, masking changes a field's modal value in 1–4 of 25 events per model and prompt in Table 10.

6 Discussion and Scope

A verifier given the corrected target samples the conditional law up to its numerical path, for two drafts and two targets and for the numerical path with Qwen3-8B, on canonical leaf-stopping tries under prompts with and without the schema, at temperature 1. These events are a subset of the valid outputs: under schema-stating prompts three layouts cover a median 0.23–0.60 of schema-valid samples, and non-canonical tokenizations make up 0.01–0.69 of event hits by model in Section G. The exact-sampler matrix recomputes the prompt at every forward on a single device without batching, so its wall clock is not serving throughput and Spec-4's advantage holds for that row interface; the one-off qualification, which evaluates every row of the event, amortizes over 200–9,600 outputs in Table 28. Unrestricted JSON conditionals and downstream utility (Tam et al., 2024) lie outside the measured events.

AI USE STATEMENT

In this work, we used generative AI tools to design and improve experiments and to implement methods. We have not used generative AI tools for qualitative and thematic data analysis, and formulating questions for surveys or interviews and transcribing recordings are not applicable to this work. Additionally, we used generative AI tools to edit the paper to improve readability and to create or edit software code. We have reviewed all AI-assisted work: AI-assisted code was tested before use, and every reported number is computed from recorded experimental outputs by analysis scripts. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The authors have read and adhere to the ICLR Code of Ethics. This work involves no human subjects, no user data and no personal information. We use publicly released model checkpoints and small toy language models, with synthetic grammar events and public JSONSchemaBench examples (Geng et al., 2025); no dataset is collected or released beyond these events, their derived probability tables and the sampled outputs. The methods change how faithfully a constrained decoder follows the model's own conditional distribution; they add no capability to the model, and their potential for misuse is that of constrained decoding in general. Faithful conditional sampling makes the constrained output reflect the model's own preferences, including its biases, rather than an artifact of masking; we make no fairness claim about those preferences.

REPRODUCIBILITY STATEMENT

Definitions, assumptions and results are stated in Section 3 and Section 4, with complete proofs in Section F. The exact samplers are specified in Section E, including the pseudocode of Spec-4 in Algorithm 1 and its exhaustive check. The models, events, prompts, precisions, seeds, sample sizes and statistical procedures are given with each experiment in Section 5 and Sections A to C; the execution and timing contracts are in Section D, and every registered decision rule with its readout is in Section G. The anonymous supplementary material contains the code and the recorded outputs listed in Table 29, with commands that recompute the reported numbers.

REFERENCES

Faruk Alpay and Bilge Senturk. Attention meets reachability: Structural equivalence and efficiency in grammar-constrained LLM decoding. *arXiv preprint arXiv:2603.05540*, 2026. URL https://arxiv.org/abs/2603.05540.

Carme Álvarez and Birgit Jenner. A very hard log-space counting class. *Theoretical Computer Science*, 107(1):3–30, 1993. doi: 10.1016/0304-3975(93)90252-O.

Emmanuel Anaya Gonzalez, Sairam Vaidya, Kanghee Park, Ruyi Ji, Taylor Berg-Kirkpatrick, and Loris D'Antoni. Constrained sampling for language models should be easy: An MCMC perspective. In *Advances in Neural Information Processing Systems*, volume 38, pp. 59951–59981, 2025. doi: 10.52202/085713-2005. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5681251fa039cf49d6d11b906eded1b3-Abstract-Conference.html.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Guiding LLMs the right way: Fast, non-invasive constrained generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3658–3673, 2024. URL https://proceedings.mlr.press/v235/beurer-kellner24a.html.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5209–5235, 2024. URL https://proceedings.mlr.press/v235/cai24b.html.

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

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.06036. arXiv:2602.06036.

Vincenzo Collura, Karim Tit, Eleonora Giunchiglia, Mike Papadakis, and Maxime Cordy. Stay within your bounds: Distance-guided decoding for guaranteed context-free grammar compliance. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. URL https://arxiv.org/abs/2608.28229. arXiv:2608.28229.

Meihua Dang, Linxin Song, Honghua Zhang, Jieyu Zhao, Guy Van den Broeck, and Stefano Ermon. Mitigating bias in locally constrained decoding via tractable proposals. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.01926. arXiv:2606.01926.

Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ziyi Xu, Yilong Zhao, Ruihang Lai, and Tianqi Chen. XGrammar: Flexible and efficient structured generation engine for large language models. In *Proceedings of Machine Learning and Systems*, volume 7, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/hash/5c20ca4b0b20b0bd2f1d839dc605e70f-Abstract-Conference.html.

Joseph L. Doob. *Classical Potential Theory and Its Probabilistic Counterpart*, volume 262 of *Grundlehren der mathematischen Wissenschaften*. Springer, New York, 1984. doi: 10.1007/978-1-4612-5208-5.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A rigorous benchmark of structured outputs for language models. *arXiv preprint arXiv:2501.10868*, 2025.

Joshua Goodman. Semiring parsing. *Computational Linguistics*, 25(4):573–606, 1999. URL https://aclanthology.org/J99-4004/.

Vivek Gore, Mark Jerrum, Sampath Kannan, Z. Sweedyk, and Stephen Mahaney. A quasi-polynomial-time algorithm for sampling words from a context-free language. *Information and Computation*, 134:59–74, 1997. doi: 10.1006/inco.1997.2621.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/bc667ac84ef58f2b5022da97a465cbab-Abstract-Conference.html.

Ilya Koziev, Leonid Sinev, and Ivan Oseledets. How lossless is lossless speculative decoding? The role of numerical precision in Orthrus. *arXiv preprint arXiv:2609.15504*, 2026. URL https://arxiv.org/abs/2609.15504.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Linzhang Li, Yixin Dong, Guanjie Wang, Ziyi Xu, Alexander Jiang, and Tianqi Chen. XGrammar-2: Dynamic and efficient structured generation engine for agentic LLMs. In *Proceedings of the ACM Conference on AI and Agentic Systems (CAIS '26)*, pp. 1009–1022. ACM, 2026. doi: 10.1145/3786335.3813124.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948, 2024. URL https://proceedings.mlr.press/v235/li24bt.html.

Benjamin Lipkin, Benjamin LeBrun, Jacob Hoover Vigly, João Loula, David R. MacIver, Li Du, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Timothy J. O'Donnell, Alexander K. Lew, and Tim Vieira. Fast controlled generation from language models with adaptive weighted rejection sampling. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=3BmPSFAdq3.

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, and Timothy J. O'Donnell. Syntactic and semantic control of large language models via sequential Monte Carlo. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a2d537e69a6c6638a3630eef835f07de-Paper-Conference.pdf.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. NeuroLogic A*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 780–799. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.57. URL https://aclanthology.org/2022.naacl-main.57/.

Kuldeep S. Meel and Alexis de Colnet. #CFG and #DNNF admit FPRAS. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 5978–6010. Society for Industrial and Applied Mathematics, 2026. doi: 10.1137/1.9781611978971.213.

Daniel Melcer, Sujan Kumar Gonugondla, Pramuditha Perera, Haifeng Qian, Wen-Hao Chiang, Yanjun Wang, Nihal Jain, Pranav Garg, Xiaofei Ma, and Anoop Deoras. Approximately aligned decoding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 11445–11479, 2025. doi: 10.52202/085713-0383. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/10beb9c6620e72264b093899a341924d-Abstract-Conference.html.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '24)*, pp. 932–949. ACM, 2024. doi: 10.1145/3620666.3651335.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, Yaguang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36486–36503, 2024. URL https://proceedings.mlr.press/v235/mudgal24a.html.

Nishanth Sridhar Nakshatri, Shamik Roy, Rajarshi Das, Suthee Chaidaroon, Leonid Boytsov, and Rashmi Gangadharaiah. Constrained decoding with speculative lookaheads. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4681–4700, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.239. URL https://aclanthology.org/2025.naacl-long.239/.

Işıl Özgü, Yaoxuan Wu, Guy Van den Broeck, and Miryung Kim. The parser already knows: Lightweight bias correction in constrained decoding. *arXiv preprint arXiv:2608.10137*, 2026. URL https://arxiv.org/abs/2608.10137.

Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D'Antoni. Grammar-aligned decoding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24547–24568, 2024. doi: 10.52202/079017-0774. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2bdc2267c3d7d01523e2e17ac0a754f3-Abstract-Conference.html.

Paweł Parys, Sairam Vaidya, Taylor Berg-Kirkpatrick, and Loris D'Antoni. Constrained adaptive rejection sampling. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62468. arXiv:2510.01902.

Ibne Farabi Shihab, Md Najmus Swaqeeb, and Abu Sa-Adat Mohamed Moon-Im Al Ahsan. Stateful CARS: Exact cross-history reuse for policy-constrained LLM agents. *arXiv preprint arXiv:2608.08282*, 2026. URL https://arxiv.org/abs/2608.08282.

Andreas Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201, 1995.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. SpecTr: Fast speculative decoding via optimal transport. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2310.15141.

Ziteng Sun, Uri Mendlovic, Yaniv Leviathan, Asaf Aharoni, Jae Hun Ro, Ahmad Beirami, and Ananda Theertha Suresh. Block verification accelerates speculative decoding. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3e710b42b1a9ed898f607ec0f4fcc971-Abstract-Conference.html.

Killian Susini and Tristan Cazenave. Improving grammar constraints generation alignment by sampling highly probable playouts. In *AAAI 2026 Workshop on Next-Gen Code Development with Collaborative AI Agents*, 2026. URL https://www.lamsade.dauphine.fr/~cazenave/papers/GBFSGS_rework.pdf.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1218–1236, 2024. URL https://aclanthology.org/2024.emnlp-industry.91/.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J. Weinberger. Inequalities for the L1 deviation of the empirical distribution. Technical Report HPL-2003-97R1, Hewlett-Packard Laboratories, 2003.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*, 2023. URL https://arxiv.org/abs/2307.09702.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, 2021. URL https://aclanthology.org/2021.naacl-main.276/.

Haotian Ye, Himanshu Jain, Chong You, Ananda Theertha Suresh, Haowei Lin, James Zou, and Felix Yu. Efficient and asymptotically unbiased constrained decoding for large language models. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4483–4491, 2025. URL https://proceedings.mlr.press/v258/ye25b.html.

Gwen Yidou-Weng, Ian Li, Anji Liu, Oliver Broadrick, Yuchen Cui, Guy Van den Broeck, and Benjie Wang. Learning tractable distributions of language model continuations. *arXiv preprint arXiv:2511.16054*, 2025. URL https://arxiv.org/abs/2511.16054.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 188045–188077, 2025. doi: 10.52202/085713-5653.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60704–60748, 2024. URL https://proceedings.mlr.press/v235/zhao24c.html.

A Exact References, Numerical Paths and Tokenization

Why these laws are exact. On a finite trie, a sampler whose next-token law depends only on the current node has terminal law $\prod_t c(v_t)(x_t)$, the product of its node conditionals along the path. We therefore compute, rather than estimate, the terminal laws of local masking (N); stepwise rejection (W) as in the sampling step of AWRS (Lipkin et al., 2025) without its SMC correction, where redrawing from the full vocabulary until the token is admissible has exactly the masked law; global rejection (R); corrected autoregressive decoding (S); and direct sampling from the enumerated terminal law (T). The tables are the fixed references of the numerical-path study, in fp32 and bf16 for A and B and in bf16 for C–E. A check guards the machinery in both directions: on a hand-built trie an exact sampler is measured at TV 1.4×10^{-3} from 200,000 draws, inside the plug-in 99% envelope 2.5×10^{-3}, while the same measurement applied to the masked sampler returns 0.0521 against its hand-computed gap 0.0519.

Table 5: Exact terminal-law baselines on the fixed tables. n_{nt} is the number of non-terminal trie prefixes, each costing one table forward, and d the string depth, which is the number of forwards per corrected autoregressive sample. TV(W)=TV(N); TV(R)=TV(S)=TV(T)$= 0$, and T needs no forward per sample. R attempts are the expected attempts per accepted string, $1/\Phi_{\mathrm{root}}$. The ASAp column gives the mean number of samples over 20 seeds, all of which converged, until its estimate equals the exact law, and the corresponding forwards as a multiple of n_{nt} under fresh-forward accounting.

Schema	dtype	Strings	n_{nt}	d	TV(N)	R attempts	ASAp samples (forwards)
A	fp32	3	7	5	0.193	1.8×10^{8}	3.20 ($2.3\, n_{\mathrm{nt}}$)
A	bf16	3	7	5	0.214	1.5×10^{8}	3.20 ($2.3\, n_{\mathrm{nt}}$)
B	fp32	4	16	9	0.428	1.5×10^{11}	9.85 ($5.5\, n_{\mathrm{nt}}$)
B	bf16	4	16	9	0.435	2.0×10^{11}	8.75 ($4.9\, n_{\mathrm{nt}}$)
C	bf16	18	70	13	0.174	1.7×10^{19}	—
D	bf16	24	104	14	0.681	4.0×10^{17}	—
E	bf16	2,000	2,232	11	0.244	2.4×10^{12}	—

Reading the table. Masking and stepwise rejection carry a fixed bias that no sample size removes. Global rejection is exact but infeasible: the target assigns only 6×10^{-20} to 7×10^{-9} probability to the valid set. Corrected autoregressive decoding pays the n_{nt} table forwards once and d forwards per sample, so at $M = 10{,}000$ samples its forward overhead over masking, $n_{\mathrm{nt}}/(Md)$, is between 0.01% on A and 2.0% on E. The CARS arm of Section 5.2 is specified in Section E.2.

ASAp and the GAD code release. ASAp keeps an over-approximation of expected future grammaticality, initialized to one and refreshed bottom-up after each sample (Park et al., 2024). Its soundness theorem requires an over-approximation, and one is the only constant that is sound at every prefix, so we use initial value one. We ran the unmodified estimator file of the public GAD code release (Park et al., 2024) with our canonical trie as the grammar engine: over 5 seeds and 120 samples per cell on A and B in both dtypes it produced the same sample sequences as our reimplementation, with final-law TV differences of 6.2×10^{-9} to 3.0×10^{-8}, consistent with float32 versus float64 arithmetic; a script in the supplement reruns this comparison. The GAD constrained-decoding front end supports byte-level tokenizers only through its GPT-2 class, so we use the ASAp estimator with our trie as the grammar engine.

Twenty unseen schemas. The 20 schemas of Section 5.3 were generated from structural rules before any model call on them: five each of enumerations, nesting, optional fields and field dependencies, excluding the key names of A–E, with 829 strings and 3,168 trie nodes in total and at most 444 nodes per schema. Table 6 gives each schema's prompt-mean masking gap on Qwen3-8B fp32 canonical tables; the native and canonical fp32 reference laws agree to TV at most 2.4×10^{-5}, with median 9.9×10^{-6}.

The same events with and without the schema in the prompt. The registered schema-stating test of Table 30 rebuilds the canonical tables of the 25 events A–E and U01–U20 for each model under

Table 6: Masking gap $\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^{\star})$ of the twenty registered unseen schemas, as the mean over three prompts on Qwen3-8B fp32 canonical tables, and the TV between the native and canonical fp32 reference laws.

Schema	Gap	Native vs. canonical	Schema	Gap	Native vs. canonical
U01	0.459	1.7×10^{-5}	U11	0.990	8.9×10^{-6}
U02	0.233	2.2×10^{-5}	U12	0.998	4.9×10^{-6}
U03	0.115	1.6×10^{-5}	U13	0.993	2.3×10^{-6}
U04	0.770	9.4×10^{-6}	U14	0.969	1.5×10^{-5}
U05	0.742	1.5×10^{-5}	U15	0.990	2.8×10^{-6}
U06	0.413	1.1×10^{-5}	U16	0.685	1.8×10^{-5}
U07	0.319	1.5×10^{-5}	U17	0.471	1.0×10^{-5}
U08	0.454	2.4×10^{-5}	U18	0.359	1.6×10^{-5}
U09	0.385	2.2×10^{-5}	U19	0.385	3.0×10^{-6}
U10	0.669	1.6×10^{-5}	U20	0.391	1.7×10^{-5}

three raw prompts at temperature 1: P_J, the instruction "Output a valid JSON object:" followed by a line break, which does not state the schema; P_S, which gives the compact JSON Schema and asks for a single compact JSON object with the properties in the listed order; and P_N, which lists the properties in natural language and asks for one compact JSON object with exactly these properties in this order. Within a model, the event, the tokenizer-specific canonical trie and the leaf-stopping rule are fixed, so the prompt is the only change. Table 7 summarizes the gaps; the paired reduction is the registered secondary endpoint, the schema-level median of $\mathrm{gap}(P_J) - \mathrm{gap}(P)$ with its schema-bootstrap 95% interval.

Table 7: Masking gap $\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^{\star})$ on the same 25 events under the prompt P_J, which does not state the schema, and the prompts P_S and P_N, which do, on fp32 canonical tables. Columns: median Φ_{root}; median gap with range; number of the 25 events with gap ≥ 0.02 and ≥ 0.10; number with a smaller gap than under P_J; paired reduction with its 95% interval.

Model	Prompt	Φ_{root}	Gap [range]	≥ 0.02	≥ 0.10	$< P_J$	Paired reduction
Qwen3-8B	P_J	1.8×10^{-18}	0.476 [0.057, 0.998]	25	24	—	—
	P_S	0.151	0.051 [0.020, 0.406]	25	4	23	0.358 [0.172, 0.590]
	P_N	0.313	0.029 [0.005, 0.211]	19	2	25	0.464 [0.186, 0.615]
Qwen3.5-9B	P_J	1.9×10^{-13}	0.277 [0.017, 0.881]	24	24	—	—
	P_S	0.032	0.029 [0.002, 0.468]	16	4	22	0.234 [0.153, 0.383]
	P_N	0.127	0.009 [0.001, 0.372]	9	2	23	0.272 [0.187, 0.444]

Gap, rarity and dispersion from the node tables. The six node tables of the panel, which list every non-terminal prefix with the target's log-probabilities of its legal children, determine Φ at every node by backward summation, and hence $\mu^{\star}$, μ^{proj}, the path weights W of Proposition 5 and $D = D_{\mathrm{KL}}(\mu^{\star} \| \mu^{\mathrm{proj}})$. Recomputed this way, the 150 gaps and root survival probabilities equal the archived readings, with largest differences of 2.7×10^{-16} in gap and 0 in Φ_{root}; the identity $\mathrm{TV} = \frac{1}{2}\mathbb{E}_{\mu^{\mathrm{proj}}}|W/Z_{\mathcal{C}} - 1|$ holds to a relative 4×10^{-14}, and no instance exceeds $1 - \Phi_{\mathrm{root}}$ or the Pinsker bound $\sqrt{D/2}$. The association analysis is a secondary analysis of these readings, registered after the archived gaps and root masses were recorded and before any correlation was computed: over the 150 instances, Spearman's ρ between gap and $-\log \Phi_{\mathrm{root}}$ is 0.66, with a 95% interval of 0.59–0.72 from 10,000 bootstrap resamples of the 25 events that use average ranks for the ties resampling creates, and between gap and D it is 0.99. The association runs mainly between prompts; within a model–prompt group the gap follows D, and the within-group rarity correlations are in Table 30. Table 8 lists the groups.

Three acquisition tasks. The paper uses target rows for three different purposes, and their costs are not interchangeable. The exact-Φ verifier target needs the full table, one row per non-terminal prefix per prompt by Proposition 7, after which any number of corrected samples costs only the

Table 8: Median gap, root survival and visit-weighted dispersion $D = D_{\mathrm{KL}}(\mu^\star \| \mu^{\mathrm{proj}})$ per model and prompt over 25 events each, recomputed from the node tables; the last column is Spearman ρ between gap and D over the 25 events.

Model	Prompt	$\log_{10} \Phi_{\mathrm{root}}$	Gap	D	$\rho(\mathrm{gap}, D)$
Qwen3-8B	P_J	-17.8	0.476	0.629	0.985
	P_S	-0.82	0.051	8.2×10^{-3}	0.935
	P_N	-0.50	0.029	3.3×10^{-3}	0.965
Qwen3.5-9B	P_J	-12.7	0.277	0.297	0.902
	P_S	-1.49	0.029	2.5×10^{-3}	0.989
	P_N	-0.89	0.009	2.3×10^{-4}	0.990

verifier's own forwards. An exact sampler such as CARS or Spec-4 needs no table and no certificate: it acquires rows adaptively, and rows acquired for one output are reused by later outputs of the same history. A certified approximate law from a query budget, as in Propositions 8 and 9, needs rows until the certificate reaches the required level; Proposition 9 bounds what any certificate can say for a given set of rows and says nothing about the order in which they were queried. Table 9 reports the second and third costs on the same instances.

Exact cost and certification on the same instances. Two further readouts on the same 150 instances were registered as descriptive analyses before they were computed. First, the recorded CARS sampler of Section 5.2, with its code unchanged, runs on a row backend that returns each node table's legal-child probabilities, renormalizing the 43 rows whose probabilities sum above one by rounding; for 1 and 16 outputs per history and ten sampling seeds per instance we count the target rows it acquires. By Section E.3, every output is an exact draw from the table's $\mu^\star$. Second, mass-first querying with the resolved-leaf certificate is replayed on every table, recording after each query the certificate $R/(L+R)$, the bound of Proposition 9 and the actual TV of q_{leaf}. Table 9 lists both readouts. With the schema stated, one exact output needs a median 0.15–0.33 of the full table's rows after a median of at most 2.2 invalid attempts; over the 150 instances the row fraction rises with the gap, with Spearman 0.43. No step of the 150 traces has an actual TV above the certificate, and at every step with $L > 0$ the certificate is at most twice the bound of Proposition 9; before a valid leaf is resolved, neither q_{leaf} nor the bound is defined. With the schema stated, the median actual TV at first certification of 0.02 is 0.003–0.017, within a factor of seven of the certified level.

Table 9: Exact acquisition cost and certification on the 25-event prompt panel, from fp32 node tables, as medians over the 25 events of each group with the interquartile range in brackets. CARS: target rows for one exact output as a fraction of the full exact table, that is of its nonterminal prefixes; rows per output for 16 outputs on one persistent cache; invalid attempts for one output, averaged over ten sampling seeds per instance. Certificate: fraction of the table's queries at which mass-first querying first certifies $R/(L+R) \le 0.02$, and the actual TV of q_{leaf} there, given as the median followed by the maximum in parentheses.

			CARS target rows			Certificate at 0.02	
Model	Prompt	Gap	1 output / table	per output at 16	invalid	query fraction	actual TV
Qwen3-8B	P_J	0.476	0.65 [0.42, 0.89]	4.0	30.2	0.93 [0.85, 0.99]	9.3×10^{-5} (0.0013)
	P_S	0.051	0.33 [0.24, 0.49]	2.9	2.2	0.89 [0.85, 0.94]	0.0029 (0.0089)
	P_N	0.029	0.24 [0.14, 0.33]	2.5	0.9	0.79 [0.70, 0.90]	0.010 (0.016)
Qwen3.5-9B	P_J	0.277	0.56 [0.32, 0.66]	3.8	22.3	0.92 [0.67, 0.97]	0.0017 (0.0086)
	P_S	0.029	0.18 [0.12, 0.27]	3.4	0.7	0.97 [0.95, 1.00]	0.015 (0.019)
	P_N	0.009	0.15 [0.10, 0.22]	3.3	0.1	0.96 [0.90, 0.99]	0.017 (0.019)

What the gap changes in the record. A further descriptive readout, registered before it was computed, asks how the gap shows up in the content of a generated record. Every terminal string of the 150 panel tables is parsed and flattened to pairs of a field path and its value; for every field–value event $\{x : \text{value at path} = v\}$ we compare its probability under μ_{proj} and $\mu^\star$, and we record whether

the most probable value of some field differs between the two laws, as listed in Table 10. The largest field–value error is close to the whole gap, with median ratio 0.96 over the 150 instances and Spearman 0.99, so the distortion is concentrated in single fields rather than spread thinly over many. With the schema stated it changes the most probable value of a field in 1–4 of the 25 events per model and prompt: for Qwen3-8B under P_S on U19, local masking makes dev the most likely value of the env field, at 0.54, whereas the conditional law puts 0.86 on staging.

Table 10: Field–value outcomes of local masking on the prompt panel, from fp32 node tables with 25 events per group. $\mathrm{FV}_{\max}$: largest $|\mu_{\text{proj}}(e) - \mu^\star(e)|$ over the event's field–value events e; flipped: events in which the most probable value of at least one field differs between μ_{proj} and $\mu^\star$; largest: the group's largest $\mathrm{FV}_{\max}$, followed by its event and the field's path in parentheses.

Model	Prompt	median $\mathrm{FV}_{\max}$	$\mathrm{FV}_{\max} \geq 0.10$	flipped	largest (event, field)
Qwen3-8B	P_J	0.475	22	16	0.996 (U12, `/format`)
	P_S	0.042	3	4	0.406 (U19, `/env`)
	P_N	0.026	2	1	0.211 (U07, `/config/kind`)
Qwen3.5-9B	P_J	0.234	22	11	0.873 (U12, `/size`)
	P_S	0.023	4	4	0.468 (U06, `/config/level`)
	P_N	0.009	2	2	0.372 (U19, `/env`)

Tokenization multiplicity. A dynamic program over byte positions with the Qwen3 vocabulary counts the token sequences that spell each string: 1,040–27,040 on A, 7,888–11,832 on B, at least 10^7, the counting cap, on C and D, and 51,840–112,320 on 24 sampled E strings. Our tries carry exactly one, the canonical tokenization. A character-level constrained decoder conditions on the union of these tokenizations, so its terminal law is compared with our $\mu^\star$ only after one target is rewritten onto the other's conditioning event.

How far each precision is from the next. We rebuilt all three precisions for A–E in one session on the same hardware, from one copy of the weights, so that only the forward precision differs in Table 11; in every rung the final logits are cast to fp32 before the log-softmax, so the top rung is an fp64 forward with fp32 normalization. Moving the forward from fp32 to fp64 changes the reference law by at most 3.0×10^{-6}; moving bf16 to fp32 changes it by 0.008 to 0.133, four to five orders of magnitude more. The headline corrected residual, 0.0072, is about $2{,}400$ times the fp64–fp32 forward disagreement. Rebuilding on other hardware from a different copy of the weights, the fp32 laws agree to 2.7×10^{-6}, while the bf16 laws differ by 0.020 to 0.194. We therefore report fidelity on fp32 and treat bf16 reference values as specific to the hardware that produced them.

Table 11: Distance between adjacent precisions. Columns 2–3 are rebuilt in one session on the same hardware from one copy of the weights; columns 4–5 compare these tables with the fixed tables built in a separate session on other hardware at the same precision. The stored reference contains fp32 only for A and B. All values are exact terminal-law TV, not sampled. Subscripts give the forward precision; the log-softmax is computed in fp32 in every rung.

	one session, same hardware		across hardware, same dtype	
Schema	$\mathrm{TV}(\mu^\star_{64}, \mu^\star_{32})$	$\mathrm{TV}(\mu^\star_{32}, \mu^\star_{16})$	fp32	bf16
A	2.1×10^{-7}	0.0089	2.7×10^{-6}	0.020
B	5.6×10^{-7}	0.0078	2.7×10^{-6}	0.050
C	1.6×10^{-6}	0.0246	—	0.036
D	3.0×10^{-6}	0.1333	—	0.194
E	9.8×10^{-7}	0.0201	—	0.024

Three further seeds and an independent rerun. Three further fp32 native runs with three pre-registered seeds, $N = 10{,}000$ per cell and criteria fixed before the first run, give corrected TV to $\mu^\star$ of 0.0012–0.0334 on A–E, with A–D means of 0.0053, 0.0085 and 0.0078 against 0.0072 for seed 42, and uncorrected TV of 0.185–0.617 in Table 12. None of the 15 corrected cells is rejected

under Holm; the smallest adjusted p in the per-seed ten-cell families is 0.22. All fifteen reference tables rebuilt for these runs are identical to those of the seed-42 run, the same code having evaluated each of the five tries four times in separate runs. An independent rerun of seed 42 on other hardware with identical weights reproduces all 20 fp32 count vectors of the five schemas bit for bit, native and canonical paths, both arms, and the reference laws agree to the last digit, so every fp32 count vector is the same on both machines. The table runs of the subsequent studies behave the same way: each of 35 independent repeats, launched with the prediction that it would match its primary bit for bit, reproduces its table log-probabilities, instance fields and probe counts exactly, 2,720,039 table log-probabilities in total across the 35 repeats.

Table 12: Corrected-arm TV to $\mu^\star$ in the seed-42 run and the three pre-registered fp32 seeds, the pointwise 99% multinomial quantile of the run's own reference, the smallest Holm-adjusted p over the three seeds, and whether every fp32 count vector of the seed-42 run is bitwise identical in the independent rerun on other hardware, with the number of cells compared. The last two rows give the A–D means of both arms. †: above its pointwise quantile.

Schema	seed 42	corrected TV to $\mu^\star$, seeds 1–3			99% q.	min Holm p	independent rerun bitwise (cells)
A	0.0062	0.0012	0.0056	0.0030	0.0083	0.22	yes (4)
B	0.0012	0.0025	0.0074	0.0122	0.0125	0.58	yes (4)
C	0.0082	0.0082	0.0044	0.0090	0.0163	1.00	yes (4)
D	0.0134	0.0094	0.0166†	0.0070	0.0165	0.75	yes (4)
E	0.0269	0.0323	0.0334	0.0318	0.0377	1.00	yes (4)
Mean A–D, corrected	0.0072	0.0053	0.0085	0.0078	—	—	—
Mean A–D, uncorrected	0.355	0.356	0.355	0.356	—	—	—

The 40-cell numerical-path family. Table 14 lists every cell. All four rejections and all seven cells outside their pointwise 99% TV envelopes are native bf16 cells. The node-level native/canonical kernel difference of the corrected arm reaches 0.0085, 0.054, 0.029, 0.024 and 0.064 on A–E in bf16 and at most 1.7×10^{-5} in fp32, and its accept/reject arithmetic error is at most 1.0×10^{-7}. Within a session the A and B reference tables are bitwise identical across four runs, and the fp32 tables of all five schemas are identical to those rebuilt in three separate runs on the same hardware, as Table 12 records. The adjusted p-values reproduce exactly from the supplied count vectors and reference probabilities with the original calibration of 19,999 multinomial simulations per reference contract.

Kernel replay of the rejected cells. The verifier logs, for every visited node and verification stage, the masked target kernel it actually used. A registered replay forms, for each node, the visit-count-weighted mean of these native kernels, multiplies them along each string to a replay law μ_{rep}, and compares three distances: $\mathrm{TV}_{\mathrm{ref}} = \mathrm{TV}(\hat{\mu}_{\mathrm{obs}}, \mu_{\mathrm{ref}})$ to the same-dtype canonical reference, $\mathrm{TV}_{\mathrm{rep}} = \mathrm{TV}(\hat{\mu}_{\mathrm{obs}}, \mu_{\mathrm{rep}})$, and $\mathrm{TV}_{\mathrm{shift}} = \mathrm{TV}(\mu_{\mathrm{rep}}, \mu_{\mathrm{ref}})$. The explained fraction is

$$\mathrm{EF} = 1 - \mathrm{TV}_{\mathrm{rep}}/\mathrm{TV}_{\mathrm{ref}},$$

and a cell is explained if $\mathrm{EF} \geq 0.80$ and a Monte Carlo Pearson test of the counts against μ_{rep} with 19,999 draws has $p \geq 0.01$. Table 13 lists the four rejected cells: the replay accounts for at least 80% of the residual in the cells on A and B and for 67% on C. The replay averages each node's kernel over visits, so variation between visits, whose block position and cache state differ, is not represented; the spread column reports it. As a registered specificity control, no canonical cell, whose verifier does not use the logged native kernels, combines a high EF with a large shift.

Table 13: Registered kernel replay of the four Holm-rejected bf16 native cells. Spread is the largest within-node range of the logged kernel over visits.

Try	Arm	$\mathrm{TV}_{\mathrm{ref}}$	$\mathrm{TV}_{\mathrm{rep}}$	$\mathrm{TV}_{\mathrm{shift}}$	EF	Replay p	Spread
A	masked	0.0250	0.0007	0.0251	0.973	0.76	0
A	corrected	0.0106	0.0020	0.0085	0.807	0.51	0
B	corrected	0.0379	0.0006	0.0373	0.985	0.99	0.026
C	masked	0.0236	0.0078	0.0192	0.671	0.36	0.013

Block length. On schema B with the same target and draft, the bf16 native path has terminal TV 0.02718, 0.03568, 0.03031 and 0.03787 at block lengths 2, 4, 8 and 16, all above the same-dtype 99% multinomial envelope 0.01292, while the canonical path gives 0.00347, 0.00653, 0.00288 and 0.00707. The fp32 comparison gives 0.00119 for both paths with identical counts and a 99% envelope of 0.01248. The bf16 node-level native/canonical kernel discrepancy reaches TV 0.05359, the fp32 maximum approximately 1.2×10^{-6}: the bf16 residual follows the block forward at every block length, and the fp32 path matches its reference.

Table 14: 40-cell numerical-path family: Qwen3-8B target, z-lab DFlash-b16 draft, standard accept/reject, schemas A–E, $N = 10{,}000$ terminal draws per cell, one Holm family. Entries are TV to each arm's own null—the canonical masked law for SchemaTV and the canonical corrected law for exact-Φ—with the 99% quantile of 19,999 multinomial simulations under that reference in parentheses; * marks a Holm-adjusted Monte Carlo Pearson $p < 0.05$. The canonical path gives the verifier the per-prefix reference probabilities; the native path uses its own cached block forward. In fp32 the two paths return identical counts.

Schema	dtype	Path	Uncorrected, SchemaTV	Corrected, exact Φ
A	fp32	native, canonical	0.0019 (0.0121)	0.0062 (0.0083)
A	bf16	native	0.0250 (0.0123)*	0.0106 (0.0080)*
A	bf16	canonical	0.0015 (0.0123)	0.0037 (0.0080)
B	fp32	native, canonical	0.0021 (0.0143)	0.0012 (0.0125)
B	bf16	native	0.0063 (0.0145)	0.0379 (0.0129)*
B	bf16	canonical	0.0083 (0.0145)	0.0071 (0.0129)
C	fp32	native, canonical	0.0128 (0.0190)	0.0082 (0.0163)
C	bf16	native	0.0236 (0.0189)*	0.0166 (0.0160)
C	bf16	canonical	0.0095 (0.0189)	0.0109 (0.0160)
D	fp32	native, canonical	0.0062 (0.0153)	0.0134 (0.0165)
D	bf16	native	0.0134 (0.0151)	0.0102 (0.0165)
D	bf16	canonical	0.0046 (0.0151)	0.0076 (0.0165)
E	fp32	native, canonical	0.0230 (0.0305)	0.0269 (0.0377)
E	bf16	native	0.0469 (0.0297)	0.0469 (0.0369)
E	bf16	canonical	0.0245 (0.0297)	0.0293 (0.0369)

A.1 HEAD/TAIL BOUND ON THE FULL TERMINAL LAW

This post hoc bound was defined after the registered bounds of Table 3 were known; its grid and δ were fixed before it was computed.

Setting. Let K and $\mu^\star$ be laws on k atoms, H a set of h atoms chosen from $\mu^\star$ alone and T its complement. Let TV_c be the TV between the coarsened laws on the $h+1$ cells formed by each atom of H and the lumped T.

The bound. The tail atoms satisfy

$$\sum_{i \in T} |K_i - \mu^\star_i| \le K(T) + \mu^\star(T),$$

and the lumped cell contributes

$$\tfrac{1}{2}\,|K(T) - \mu^\star(T)|$$

to TV_c. Hence

$$\begin{aligned} \mathrm{TV}(K, \mu^\star) &\le \mathrm{TV}_c(K, \mu^\star) + \min\{K(T), \mu^\star(T)\} \\ &\le \mathrm{TV}_c(\hat{p}, \mu^\star) + \mu^\star(T) + \varepsilon_{h+1}, \end{aligned}$$

where the last step applies the Weissman inequality to the coarsened law with the $N = 40{,}000$ pooled draws:

$$\varepsilon_{h+1} = \sqrt{\frac{\ln(2^{h+1} - 2) + \ln(G/\delta)}{2N}}.$$

Grid and values. H holds the h atoms of largest $\mu^\star$ mass, $h \in \{1, 2, 4, \ldots, 1024\}$ below k plus $h = k$, which has no tail; G is the grid size and $\delta = 0.05/5$ per try, as for the registered bounds; the minimum over the grid is valid by the Bonferroni split. On E, where $G = 12$, the minimum is at $h = 64$: $\mu^\star(T) = 0.0241$, $\varepsilon_{65} = 0.0255$ and bound 0.0581, or 0.059 rounded up, against a masking gap of 0.2306; $h = k$ gives 0.149. On A–D the minima are 0.0131, 0.0124, 0.0202 and 0.0201, of the same order as the registered bounds, which Table 3 keeps. The script and its output are in the supplement.

B A Second Verifier with a Chain Draft

Design. The second implementation and its sample sizes were registered before its first run; the registration is in the supplement. It keeps the Qwen3-8B target, prompt and tries A–D of Section 5.1 and replaces the block draft by the autoregressive bf16 Qwen3-1.7B draft, whose next-token law is restricted to the trie children of the drafted prefix. Each round drafts four tokens; the target scores them in one KV-cached forward; position j is accepted with probability $\min\{1, p_j/q_j\}$ and otherwise replaced by a draw from the normalized residual $(p-q)^+$, and a bonus token is drawn from the target when all four are accepted. The reference $\mu^\star$ is the backward-Φ law over no-cache single-prefix target rows in the run's dtype. "Native" cells give the rule the target rows of the KV-cached verify forward, "canonical" cells the reference rows; masked cells use μ_t^{proj} and corrected cells $\mu_t^\star$. Target dtype, fp32 or bf16, is the manipulated factor. Sample sizes are $N = 10{,}000$ for the fp32 native corrected cells and all bf16 corrected cells and $N = 2{,}000$ for the masked and fp32 canonical cells; all 24 cells completed. Three families were registered: P1, the fp32 native masked cells against μ^{proj} with Holm over 4; P2, the fp32 native corrected cells against $\mu^\star$ with Holm over 4 and the Weissman bound of Section 5.1 at $\delta = 0.05/4$ per try; P3, the bf16 corrected cells, native against canonical with Holm over 8, together with the node-level native/canonical row difference. Tests use Pearson statistics with 19,999 multinomial null draws.

Results. Table 15 lists every cell. P2: the fp32 corrected arm has TV 0.0013–0.0081 to $\mu^\star$, no cell is Holm-rejected, the smallest adjusted p being 0.75, and the simultaneous 95% bounds, rounded up, of 0.019, 0.022, 0.036 and 0.041 remove at least 80% of the masking gaps 0.193, 0.428, 0.187 and 0.616. P1: the masked arm is within 0.008–0.015 TV of μ^{proj}, inside the 95% quantile of its own null on every try, and no cell is rejected, so the masking gap is a property of the supplied target and not of the draft. P3: the largest node-level native/canonical row difference over the family's visited nodes is 1.6×10^{-5} in fp32 and 0.028 in bf16; per try it is larger in bf16 on A, C and D, and below 5×10^{-6} in both dtypes on B. The one Holm rejection among the eight bf16 corrected cells is a native cell, on A, and no canonical cell is rejected. Per-position acceptance of the fp32 corrected arm is 0.86, 0.94, 0.88 and 0.92 on A–D.

Table 15: Registered second verifier: Qwen3-8B target, Qwen3-1.7B chain draft with four tokens per round, tries A–D. Entries are TV to $\mu^\star$ of the run's own dtype; gap is $\mathrm{TV}(\mu^{\text{proj}}, \mu^\star)$ of the fp32 reference; the fp32 native corrected column gives the simultaneous 95% bound, rounded up, in parentheses. * marks a Holm-adjusted rejection in its registered family.

			fp32			bf16		
Try	Atoms	Gap	native corr. (95% bound)	native masked	canon. corr.	native corr.	canon. corr.	native masked
A	3	0.193	0.0013 (0.019)	0.1850	0.0007	0.0099*	0.0019	0.1762
B	4	0.428	0.0030 (0.022)	0.4136	0.0185	0.0023	0.0020	0.4204
C	18	0.187	0.0068 (0.036)	0.1787	0.0315	0.0087	0.0117	0.1717
D	24	0.616	0.0081 (0.041)	0.6148	0.0247	0.0199	0.0073	0.7011

B.1 A second target on cross-field events

Design. This extension was registered before its first run. It keeps the chain-draft verifier above, with its bf16 Qwen3-1.7B draft, four tokens per round, KV-cached verify forward and native target rows, and manipulates the target: Qwen3-14B and, for comparison, Qwen3-8B, both in fp32, with the prompt "Output a valid JSON object:" and leaf stopping. The four events have cross-field constraints: F, with 10 atoms, a city that belongs to its country among 4 countries; G, with 15 atoms, two hours with start before end; H, with 25 atoms, $a, b \in \{0, \ldots, 4\}$ and a sum field equal to $a + b$; I, with

12 atoms, a nested user object whose level range depends on the role, with a Boolean flag. Each target–event pair has a corrected cell with $N = 8{,}000$ and a masked cell with $N = 1{,}000$. Registered decisions: P1, masked cells against μ^{proj} with Holm over 8; P2, corrected cells against $\mu^\star$ with Holm over 8, the Weissman bound at $\delta = 0.05/8$ per cell, and a cell counts as supported if it is not rejected and its bound is below half its gap; cells with gap below 0.05 are reported but not counted. The families are fixed at eight cells, and a cell without a record enters them with $p = 1$.

Results. Table 16 lists all eight registered cells, and the registered analysis is final. No masked or corrected cell is Holm-rejected; the smallest adjusted p is 0.40 for the masked and 1.00 for the corrected cells. All six counted cells are supported, three on each target. In the counted cells the corrected law is within 0.034–0.050 of $\mu^\star$, removing at least 82% of gaps 0.19–0.85. On F the model nearly respects the constraint on its own and the gap is 0.021–0.032, below the registered resolution. The events are rare under this prompt, with Φ_{root} from $10^{-19.0}$ to $10^{-9.8}$. The node-level difference between the verify-forward rows and single-prefix reference rows is at most 9.1×10^{-6} TV in every cell and at most 8.1×10^{-6} in the corrected cells. Correcting the target lowers per-position acceptance by at most 0.08, from 0.989 masked to 0.910 corrected for H on Qwen3-14B.

Table 16: Registered second-target extension: chain-draft verifier, fp32 targets, events F–I. Gap is $\mathrm{TV}(\mu^{\text{proj}}, \mu^\star)$; corrected entries are TV to $\mu^\star$ at $N = 8{,}000$ with the 95% bound, Bonferroni over 8 cells and rounded up, in parentheses; masked entries are TV to μ^{proj} at $N = 1{,}000$; acceptance is per position, corrected/masked. "Removed" is $1 - \text{bound}/\text{gap}$ for the counted cells. All eight registered cells are reported.

Target	Event	Atoms	$\log_{10} \Phi_{\text{root}}$	Gap	Corrected (bound)	Removed	Masked	Acceptance
Qwen3-14B	F	10	−11.1	0.021	0.0031 (0.031)	—	0.0101	0.995/0.993
Qwen3-14B	G	15	−11.7	0.327	0.0089 (0.041)	0.87	0.0352	0.942/0.975
Qwen3-14B	H	25	−10.7	0.852	0.0123 (0.050)	0.94	0.0094	0.910/0.989
Qwen3-14B	I	12	−19.0	0.299	0.0064 (0.036)	0.88	0.0239	0.998/0.998
Qwen3-8B	F	10	−10.5	0.032	0.0054 (0.033)	—	0.0107	0.989/0.988
Qwen3-8B	G	15	−12.4	0.377	0.0090 (0.041)	0.89	0.0493	0.955/0.995
Qwen3-8B	H	25	−9.8	0.540	0.0112 (0.049)	0.91	0.0197	0.942/0.994
Qwen3-8B	I	12	−16.1	0.188	0.0048 (0.034)	0.82	0.0423	0.995/0.994

B.2 The chain-draft verifier under a schema-stating prompt

Design. This extension was registered before its first run. It keeps the verifier of Section B.1 with the Qwen3-8B target in fp32 and replaces the prompt by the panel's schema-stating prompts of Section 5.3: P_S, the JSON Schema text, and P_N, a natural-language field list. The cells are fixed by one rule: every Qwen3-8B panel instance under P_S or P_N whose recorded masking gap is at least 0.10. This gives six cells with 6–72 atoms: U19, U09, U06 and U08 under P_S, and U07 and U19 under P_N. Each cell has a corrected arm with $N = 8{,}000$ and a masked arm with $N = 1{,}000$, with the decision rules of Section B.1 over a family of six and the Weissman bound at $\delta = 0.05/6$ per cell. A cell without a record enters its Holm family with $p = 1$ and the family is marked provisional.

Results. All six cells are complete and the registered decisions are final; Table 17 lists them. In every cell the gap recomputed in the run agrees with the recorded panel value, 0.107–0.406, to three decimals. No corrected or masked cell is Holm-rejected. The corrected arms have TV 0.0031–0.0250 to $\mu^\star$ with simultaneous 95% bounds 0.027–0.084, so the verifier removes at least 33–90% of each gap; five cells meet the registered support rule of a bound below half the gap, three of four under P_S and both under P_N. The masked arms are within 0.014–0.064 TV of μ^{proj} at $N = 1{,}000$. Per-position acceptance is 0.938–0.987 in the corrected arms. With the 83–125-token prompts the node-level difference between verify-forward and reference rows is at most 2.8×10^{-5} TV.

B.3 The schema-stating cells on the second target

Design. This extension was registered before any Qwen3-14B output on these events existed. It repeats the protocol of Section B.2 on the same six cells with the Qwen3-14B target in fp32 and the same draft. Each cell is run as two independent halves with distinct seeds, each with a corrected arm

Table 17: Registered schema-stating extension with the Qwen3-8B target in fp32 and the chain draft. Entries as in Table 16; the bound is Bonferroni over the six registered cells and rounded up; removed is $1 - \text{bound}/\text{gap}$, rounded down.

Prompt	Event	Atoms	$\log_{10} \Phi_{\text{root}}$	Gap	Corrected (bound)	Removed	Masked	Acceptance
P_S	U19	6	-0.9	0.406	0.0155 (0.040)	0.90	0.0208	0.940/0.985
P_S	U09	32	-1.5	0.276	0.0120 (0.054)	0.80	0.0386	0.953/0.933
P_S	U06	72	-2.2	0.171	0.0163 (0.075)	0.56	0.0557	0.974/0.962
P_S	U08	72	-0.8	0.125	0.0250 (0.084)	0.33	0.0637	0.987/0.981
P_N	U07	36	-1.3	0.211	0.0095 (0.053)	0.75	0.0139	0.938/0.952
P_N	U19	6	-0.2	0.107	0.0031 (0.027)	0.75	0.0499	0.973/0.963

of $N = 4{,}000$ and a masked arm of $N = 500$, and the counts of the two halves are pooled. The families, the Weissman bound at $\delta = 0.05/6$ per cell and the support rule are those of Section B.2; the masking gap of each cell is that of the Qwen3-14B reference.

Results. All twelve halves are complete and the registered analysis is final; Table 18 lists every cell. The two halves of each cell build identical reference laws. No corrected or masked cell is Holm-rejected; the smallest adjusted p is 0.10 for the corrected and 0.62 for the masked cells. On this target masking distorts the schema-stated events less, with gaps 0.014–0.150 against 0.107–0.406 on Qwen3-8B. The corrected arms have TV 0.0102–0.0240 to $\mu^\star$ with simultaneous 95% bounds 0.034–0.083. The cell with the largest gap, U09 under P_S, meets the support rule, its bound removing at least 63% of the gap of 0.150; three cells have gaps below the registered resolution of 0.05, and the readout of the remaining two counted cells is in Table 30. Per-position acceptance is 0.917–0.977 in the corrected arms.

Table 18: Registered schema-stating cells on the Qwen3-14B target in fp32 with the chain draft, pooled over the two halves of each cell. Entries as in Table 17; removed is $1 - \text{bound}/\text{gap}$, rounded down, for the counted cells whose bound is below the gap.

Prompt	Event	Atoms	$\log_{10} \Phi_{\text{root}}$	Gap	Corrected (bound)	Removed	Masked	Acceptance
P_S	U19	6	-1.5	0.048	0.0130 (0.037)	—	0.0119	0.947/0.945
P_S	U09	32	-2.4	0.150	0.0145 (0.056)	0.63	0.0495	0.970/0.964
P_S	U06	72	-2.9	0.087	0.0156 (0.075)	0.14	0.0406	0.977/0.981
P_S	U08	72	-1.5	0.068	0.0240 (0.083)	—	0.0546	0.963/0.962
P_N	U07	36	-0.2	0.032	0.0128 (0.056)	—	0.0479	0.929/0.930
P_N	U19	6	-0.3	0.014	0.0102 (0.034)	—	0.0191	0.917/0.916

C Detailed Experiments

We validate the framework on tractable grammars where exact Φ_t is computable, so that every sampler can be compared with an exact reference. Distributional experiments use ancestral sampling at temperature 1 and report plug-in TV; sample sizes are stated with each table: real-model sampler checks use $N = 50{,}000$, or $20{,}000$ for the three-seed Qwen3-8B Dyck run; the toy-LM Dyck runs $N = 10{,}000$, the toy-LM finite JSON sweep $N = 5{,}000$, and the DFlash verifier runs $N = 10{,}000$ per schema. The estimator hierarchy of Section C.4 is evaluated through exact induced laws without sampling.

C.1 Setup

Grammars. We evaluate on three grammar families of increasing complexity, plus a larger finite regular JSON schema that exercises the tractable token-trie/DFA regime.

- **Dyck** $D_{d,L}$: context-free languages with one bracket type, maximum nesting depth d, and maximum string length L. We use $D_{3,12}$ with 145 valid strings and $D_{3,16}$ with 988 valid strings, which exhibit large $\mathrm{TV}(\mu^{\text{proj}}, \mu^\star)$ due to nesting-depth constraints that create sharply non-uniform Φ_t.

Table 19: Sampler checks on real models, one row per setting of model, prompt, d and L, with 13 settings and 17 runs at temperature 1 and $N = 50{,}000$ unless stated. The prompts are P1, "A balanced bracket sequence: <"; P2, "Balanced parens:" followed by a space; and P3, "Generate a balanced bracket string:" followed by a line break. Residual is the plug-in $\mathrm{TV}(\text{empirical}, \mu^{\text{proj}})$, so it grows with support size; Gap is the analytic $\mathrm{TV}(\mu^{\text{proj}}, \mu^{\star})$ in bf16.

Model	Prompt	d	L	Support	Residual	Gap
Qwen2.5-3B	P1	3	16	988	0.0006 / 0.0010, 2 seeds	0.045
Qwen2.5-3B	P1	3	20	6,766	0.0010	0.045
Qwen3-0.6B	P1	3	16	988	0.0104 / 0.0126, 2 seeds	0.995
Qwen3-0.6B	P1	4	24	132,867	0.0306	0.995
Qwen3-8B	P1	3	16	988	0.0020	0.9976
Qwen3-8B	P2	3	16	988	0.0006	0.022
Qwen3-8B	P1	3	20	6,766	0.0012	0.9976
Qwen3-8B	P1	4	24	132,867	0.0015	0.9976
Qwen3-8B	P3, $N = 20{,}000$, 3 seeds	3	16	988	0.0018–0.0020	0.99995
Llama-3.1-8B-Instruct	P1	3	16	988	0.0028	0.988
Llama-3.1-8B-Instruct	P1	4	24	132,867	0.0031	0.988
Vicuna-7B-v1.5	P1	3	16	988	0.0052	0.093
Vicuna-7B-v1.5	P1	3	20	6,766	0.0066	0.093

- **Regex** a^+b^+: a regular language with permissive transitions, on the toy LM; its masking gap is 0.145–0.267.
- **Finite JSON**: schemas with 3–2,000 valid strings, including a regular flag-code schema whose exact Φ computation requires backward DP over a 4,232-node token trie.
- **Budget regular DFA**: fixed-length binary strings with at most K ones. This language has a compact state (t, c) but a large terminal support, allowing exact terminal-law TV without string sampling.

Models. The sampler checks use five checkpoints from four families: Qwen2.5-3B, Qwen3-0.6B, Qwen3-8B, Llama-3.1-8B-Instruct, and Vicuna-7B-v1.5. The finite-JSON toy experiments use a character bigram model, and the Dyck toy experiments a toy LM whose bracket and EOS probabilities depend on the current depth and the remaining length; for both the exact oracle is cheap; the Qwen3-8B Dyck runs below use real prefix probabilities.

Metrics. $\mathrm{TV}(\hat{\mu}, \mu^{\star})$ is plug-in from N samples for sampled runs and exact for induced laws; $\bar{\delta}_t$, δ_t and $\bar{\Phi}_t$ are the per-prefix quantities of Theorem 6.

C.2 Sampler checks on real models

Corollary 2 predicts that every LMS-class decoder produces samples from μ^{proj}. Table 19 lists every real-model Dyck run: 17 runs over 13 distinct settings of model, prompt, d and L on five checkpoints. The sampler draws the masked kernel μ_t^{proj} at every step and commits the sample, so the residual $\mathrm{TV}(\text{empirical}, \mu^{\text{proj}})$ tests the sampler and the analytic μ^{proj} computation against each other; the verifier path itself is tested by the DFlash runs. The largest residual below the 132,867-string support is 0.0126, for Qwen3-0.6B on 988 strings; the largest among the other four checkpoints is 0.0066. The toy-LM regex runs on a^+b^+ with $L \in \{12, 16, 20, 24\}$, four seeds at $L = 16$ and $N = 50{,}000$ give residuals 0.009–0.016.

Qwen3-8B Dyck. With Qwen3-8B prefix probabilities over the 988-string support of $D_{3,16}$ under prompt P3 of Table 19, three independent seeds with $N = 20{,}000$ terminal samples each give an analytic local-projection gap of 0.999948 ± 0.000002 TV, and empirical SchemaTV matches μ^{proj} within 0.0018 ± 0.0001 TV. Exact Φ samples from $\mu^{\star}$ to the numerical floor of $1.9{\times}10^{-10}$ TV, and the true-conditioned one-step variant reaches 0.032 ± 0.002 TV. The conditioning event is rare under this prompt, with $\Phi_{\text{root}} \approx 1.4{\times}10^{-8}$.

C.3 REAL-MODEL EXACT Φ ON FINITE JSON

We query Qwen3-8B for $p(y \mid \text{prompt}, x_{<t})$ at every prefix of the five finite canonical JSON tries, which have 3 to 2,000 valid strings; the fifth is a regular code schema whose exact Φ requires backward DP over a 4,232-node token trie. We compute three laws analytically: exact $\mu^\star$ by sequence likelihood, μ^{proj} by local trie masking and renormalization, and the exact-Φ law by backward dynamic programming over the trie. Table 20 reports the gaps of this bf16 snapshot, 0.174 to 0.681 TV, and exact Φ closes the gap on every schema, with maximum residual TV below 2×10^{-15} within a fixed forward pass. The distortion is qualitative as well as quantitative: on Schema B, $\mu^\star$ assigns only 1.7% mass to strings of type b, while μ^{proj} assigns 43.8%; on Schema D, $\mu^\star$ puts 55.4% on GET/metrics/v1, while μ^{proj} puts 62.6% on GET/api/v1; on Schema E, μ^{proj} assigns 12.1% to the string with flag true and code 000, while $\mu^\star$ assigns 1.0%. Under three prompt wordings, this one and two others, on Schemas A–D all 12 prompt–schema pairs close the gap, with mean projected TV 0.405, minimum projected TV 0.119, and maximum exact-Φ residual TV 1.5×10^{-15}.

Table 20: Masking distortion in the bf16 analytic snapshot of the five tries used by the runs of this appendix, under the prompt "Output a valid JSON object:" followed by a line break. The snapshot record is in the supplement; its C–E laws are also the bf16 entries of the fixed reference, while the fixed A/B tables of Table 5 come from a separate run. μ^Φ is the product of the exact Doob kernels built from the same prefix probabilities as $\mu^\star$.

Try	Strings	$\mathrm{TV}(\mu^{\text{proj}}, \mu^\star)$	$\mathrm{TV}(\mu^\Phi, \mu^\star)$	Largest distortion
A, status	3	0.188	4.3×10^{-16}	μ^{proj} overweights error, 27% vs 8%
B, type-value	4	0.421	4.8×10^{-16}	μ^{proj} overweights type=b by 26×
C, action-target	18	0.174	1.0×10^{-15}	target/async mass shifted across fields
D, method-path	24	0.681	1.3×10^{-15}	$\mu^\star$: GET/metrics; μ^{proj}: GET/api
E, flag-code	2,000	0.244	7.0×10^{-16}	μ^{proj} overweights code 000 by 12×

Corrected speculative loop. Table 21 instantiates the corrected accept/reject loop with a locally projected draft and exact token-trie Φ and compares the empirical terminal law with $\mu^\star$. On Schemas A–D the sampled law is within TV 0.0052 of $\mu^\star$, and on the 2,000-string Schema E within TV 0.0148 at $N = 50{,}000$, against a mean masking gap of 0.342.

Structural reading. In both references, the fp32 references of Table 22 and this snapshot in Table 20, the gap appears as shifts of individual field values. These are analytic consequences of the laws, not measured tool-success or factuality outcomes. On the fp32 reference laws of the headline cells, over all 1,026 field–value marginal events of A–E, the largest masking error is 0.614, for the path /metrics, which has 0.616 under $\mu^\star$ and 0.002 under μ^{proj}; in the sampled verifier the masked and corrected arms' frequencies of each try's largest event are within 0.006 of μ^{proj} and 0.008 of $\mu^\star$.

C.4 ESTIMATOR HIERARCHY WITH EXACT INDUCED LAWS

Monte Carlo fallback and rollout validity. The Monte Carlo tier needs a rule for prefixes at which no candidate has a valid rollout. The Dyck sampler weights by $p_t(y)\max\{\hat{\Phi}^{\text{mc}}_t(y), 10^{-30}\}$, which reduces that step to local projection; the finite-JSON sampler uses the uniform law on A_t whenever all weights vanish; the exact-law analysis uses the Dyck rule. A rollout counts as valid only if it stays inside the constraint and terminates inside $\mathcal{L}(\mathcal{C})$, so the Monte Carlo estimate converges to Φ.

Table 23 gives, for each estimator on the toy-LM Dyck language $D_{3,16}$, the exact TV of the law the redrawing sampler induces, the median exact TV over 50 draws of one fixed table, the median certificate of Theorem 6(c) on those fixed tables, and the expected number of target rows, that is next-token distributions of the target at one forward each without a KV cache, weighted by the visitation law of the sampler each estimator induces. Redrawing corrected rollouts at each visit reaches single-output TV 0.071 at $k = 4$ and 0.005 at $k = 64$, and the exact table reaches $\mu^\star$. All entries are computed without sampling by a script in the supplement.

Certificates. For every deterministic tier and every one of the $2 \times 7 \times 50$ MC draws on $D_{3,12}$ and $D_{3,16}$, the per-prefix bound $\mathrm{TV}(\hat{\mu}_t, \mu^\star_t) \le \bar{\delta}_t/\bar{\Phi}_t$ held at every prefix, with 0 violations over

Table 21: Finite-trie corrected loop on Qwen3-8B prefix probabilities, with draft block $\gamma = 4$ and $N = 50{,}000$ for all rows. The loop performs speculative accept/reject sampling against the Φ-reweighted target over the same token tries and bf16 snapshot as Table 20; $\hat{\mu}^{\mathrm{corr}}$ is the empirical law of the sampler.

Schema	$\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^{\star})$	$\mathrm{TV}(\hat{\mu}^{\mathrm{corr}}, \mu^{\star})$	**Accept rate**
A, status	0.188	0.0010	0.955
B, type-value	0.421	0.0017	0.932
C, action-target	0.174	0.0029	0.978
D, method-path	0.681	0.0052	0.922
E, flag-code	0.244	0.0148	0.966
Mean	0.342	0.0051	0.951

Table 22: Largest field–value masking error per try on the fp32 reference laws of Table 2, and its frequency in the $N = 10{,}000$ samples of each fp32 verifier arm in seed 42, where both paths agree.

				Exact law		DFlash verifier	
Try	Strings	$\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^{\star})$	Event	$\mu^{\star}$	μ^{proj}	masked	corrected
A	3	0.193	status = error	0.089	0.282	0.281	0.084
B	4	0.428	type = b	0.016	0.444	0.443	0.016
C	18	0.187	target = db	0.719	0.578	0.577	0.717
D	24	0.616	path = /metrics	0.616	0.002	0.002	0.624
E	2,000	0.231	code = 000	0.018	0.211	0.205	0.019

410 and 2,816 prefixes per draw on $D_{3,12}$ and $D_{3,16}$, and the sequence sum $\sum_t \mathbb{E}_{\hat{\mu}}[\mathrm{TV}_t]$ dominated the terminal TV. For the fixed MC tables on $D_{3,16}$ the median sequence certificate is 0.27–0.37 at $k \le 8$ and 0.18–0.29 at $k \ge 16$. On $D_{3,12}$, with 145 strings, local projection has TV 0.456 and corrected-rollout MC reaches 0.121/0.071/0.034/0.015/0.007 at $k = 4/8/16/32/64$.

Grammar families. Table 24 compares grammar families on the toy LMs, where $\bar{\Phi}$ is the unweighted mean of Φ over prefix–candidate pairs. On finite JSON, exact Φ reduces TV from 0.247–0.843 to at most 0.017 at $N = 5{,}000$, the sampling-noise floor for these small supports, while the exact induced law has TV $\le 10^{-14}$; on Dyck grammars the exact oracle's sampled TV of 0.014–0.018 is the plug-in floor. Rerunning the finite-JSON sweep at bigram temperatures 0.3 and 0.7 in addition to 1.0, the local-projection gap stays large, with mean TV 0.689, 0.639 and 0.623, and exact Φ stays at the sampling floor, with mean TV ≤ 0.0082.

Multi-seed robustness. Across 10 toy-LM parameter draws on $D_{3,12}$ and $D_{3,16}$ with $N = 10{,}000$ samples per draw, where $\pm$ gives the population standard deviation, Uniform TV is 0.320 ± 0.102 and 0.319 ± 0.103 and Exact TV is 0.009 ± 0.004 and 0.009 ± 0.005: every draw has a substantial gap, and the exact arm stays at its sampling floor of $\mathrm{TV} \le 0.020$ on every draw.

Large regular-state stress test. The language

$$\mathcal{L}_{n,K} = \Big\{ x \in \{0,1\}^n : \sum_t x_t \le K \Big\}$$

under an iid Bernoulli base model has a compact DFA with state (t, c) for position and ones used, while the terminal support ranges from $6.2{\times}10^5$ to $6.1{\times}10^8$ strings in Table 25. We compute TV exactly by grouping strings with $c < K$ ones and, for $c = K$, by the position where the K-th one is reached; there is no Monte Carlo sampling. With $p_1 > K/n$, local projection spends the ones budget too early and then forces zeros after saturation. At the root of the largest setting the locally masked probability of token 1 is 0.70, while the Doob-corrected probability is 0.482, because choosing 1 leaves a lower probability of completing within the budget. The backward DP took 0.35–0.94 ms across these 225–496-state automata; token-time lookup is then table access.

Table 23: Estimator hierarchy on toy-LM Dyck $D_{3,16}$ with 988 strings, LM seed 42 and root $\bar{\Phi}_t = 0.42$, recomputed without sampling. "Induced TV" is the exact terminal TV of the law the sampler realizes, with MC rollouts redrawn at every visit and k rollouts per candidate. "Fixed table" is the median exact TV over 50 estimator draws when the MC table is drawn once and reused. "Certificate" applies to the fixed tables and is the median over draws of the sum $\sum_t \mathbb{E}_{\hat{\mu}}[\min\{1, \bar{\delta}_t/\bar{\Phi}_t\}]$ of Theorem 6(c), which held at every prefix of every draw. Target rows are exact expectations under the induced visitation law: "Estimator" counts the rows the Φ estimate needs per sampler step: one per admissible non-EOS candidate for OneStep-True, and for MC one per rollout step inside the trie, with a rollout stopping at EOS, at the length cap or on leaving the trie. "Per string" adds the sampler's own row at every visited prefix. Exact needs no estimator rows once its table of 2,816 rows, one per non-terminal prefix, has been built once. If MC rollouts are not stopped on leaving the trie, the MC estimator counts are 1.20 times larger.

				Target rows	
Estimator	Induced TV	Fixed table, median	Certificate, median	Estimator/step	Per string
Uniform	0.421	—	1	0	6.9
OneStep-Cheap	0.359	—	1	0	5.5
OneStep-True	0.295	—	1	1.53	14.2
MC(k=4)	0.071	0.271	0.27	17.5	32.4
MC(k=8)	0.039	0.170	0.37	35.7	68.1
MC(k=16)	0.019	0.091	0.29	72.3	140.7
MC(k=32)	0.009	0.066	0.24	145.6	286.7
MC(k=64)	0.005	0.060	0.18	292.1	579.0
Exact	0	0	0	0	2.0

Table 24: Grammar families on the toy LMs, seed 42, with $N = 10{,}000$ for Dyck and $N = 5{,}000$ for JSON; $\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^{\star})$ is analytic. Exact Φ closes the gap to the sampling-noise floor on every tractable grammar.

Grammar	**Type**	$\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^{\star})$	$\bar{\Phi}$	**Exact TV**
JSON status, 3 str.	Finite	0.577	0.101	<0.001
JSON type-value, 4 str.	Finite	0.247	0.113	0.009
JSON action, 18 str.	Finite	0.823	0.069	0.017
JSON method-path, 24 str.	Finite	0.843	0.066	0.006
Dyck $D_{3,12}$, 145 str.	CF	0.456	0.516	0.018
Dyck $D_{3,16}$, 988 str.	CF	0.421	0.467	0.014

Depth and length. Figure 1 uses the seed-42 toy LM and the exact induced laws of Table 23. The target includes immediate EOS at depth zero with probability 0.729. Local masking moves that probability to 0.307, changing mean maximum depth from 0.369 to 1.414 and mean length from 0.996 to 5.904. Corrected MC($k = 8$), independently redrawn at every visit and integrated over its binomial outcomes, gives depth TV 0.0347 and terminal TV 0.0391.

C.5 Observable certificates in practice

Implementation check. The released checker replays all 3,950 saved acquisition orders on tries A and B and evaluates 30,576 local midpoint-kernel comparisons of Proposition 8 against the hidden exact law, with zero bound violations at tolerance 10^{-12}. These checks use ordinary floating-point arithmetic.

Certified stopping on twenty unseen schemas. We registered a test on the 60 unseen schema–prompt instances of Section 5.3, which need 22 to 317 table queries each. Mass-first, breadth-first and rollout-Bellman orders query prefixes without any prior, the terminal bound $R/(L + R)$ of Section F.6 is recorded after every query, and exact laws are computed only after all three orders have finished. No order violates the bound at any step of any of the 180 traces; the maximum of actual TV minus bound is at most 0. Where mass-first querying first certifies 0.02, the actual TV has median 6.8×10^{-5} and maximum 0.0013. The registered query-count readout is listed in Section G.

Table 25: Large regular-state stress test. Local projection is strongly biased even for compact regular languages, and exact Φ recovers the conditional law; the maximum numerical Doob-recursion residual across these runs is 2.2×10^{-16}.

Setting	**Valid strings**	**DFA states**	$\mathrm{TV}(\mu^{\mathrm{proj}}, \mu^\star)$	$\mathrm{TV}(\mu^{\Phi}, \mu^\star)$
$n = 20, K = 10, p_1 = 0.62$	616,666	231	0.670	0
$n = 22, K = 11, p_1 = 0.65$	2,449,868	276	0.755	0
$n = 24, K = 12, p_1 = 0.68$	9,740,686	325	0.836	0
$n = 24, K = 10, p_1 = 0.65$	4,540,386	275	0.884	0
$n = 24, K = 8, p_1 = 0.70$	1,271,626	225	0.961	0
$n = 26, K = 13, p_1 = 0.68$	38,754,732	378	0.851	0
$n = 28, K = 14, p_1 = 0.68$	154,276,028	435	0.864	0
$n = 30, K = 15, p_1 = 0.70$	614,429,672	496	0.909	0

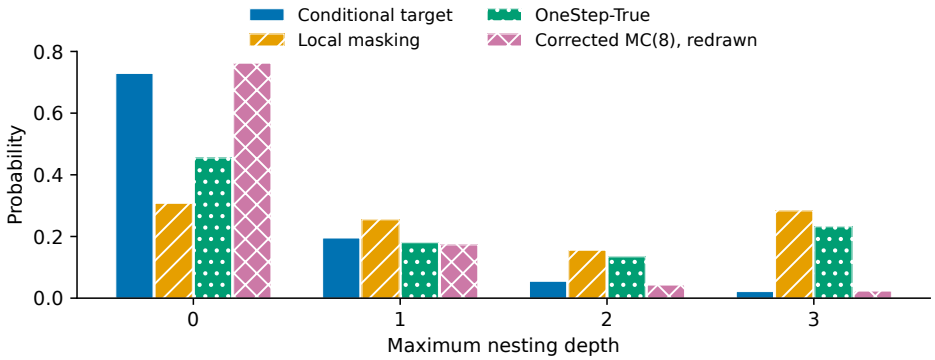

Figure 1: Exact maximum-depth marginals on Dyck $D_{3,16}$, toy LM seed 42, 988 terminal strings including EOS. MC(8) uses corrected unconstrained rollouts redrawn at every visit; the plotted law is its analytically integrated mixture, the same law as Table 23. These are probabilities, not sample histograms. Marginal TV for MC(8) is 0.0347, bounded by its terminal TV 0.0391.

Certified level, queries and actual TV. Table 26 summarizes, from the archived per-instance readings of the registered study, the fraction of the full table's queries at which each order first certifies a level and the actual terminal TV of q_{leaf} at that point. The actual TV never exceeds 0.071 times the certified level. The midpoint kernel q is a different sampler; its 30,576 local checks above use the per-step certificate b_t.

Table 26: Terminal certificate $R/(L+R)$ of the resolved-leaf sampler on the 60 registered unseen instances with Qwen3-8B fp32 tables: query fraction at first certification of each level as the median over instances with the interquartile range in brackets, and the median and maximum actual TV to $\mu^\star$ at that point.

Order	Level	Query fraction	Actual TV, median	Actual TV, max
Mass-first	0.10	0.862 [0.512, 0.976]	4.7×10^{-4}	6.9×10^{-3}
Mass-first	0.05	0.908 [0.565, 0.985]	2.0×10^{-4}	3.6×10^{-3}
Mass-first	0.02	0.934 [0.657, 0.998]	6.8×10^{-5}	1.3×10^{-3}
Breadth-first	0.10	0.952 [0.599, 1.000]	1.4×10^{-4}	5.5×10^{-3}
Breadth-first	0.05	0.964 [0.738, 1.000]	8.5×10^{-5}	2.9×10^{-3}
Breadth-first	0.02	0.976 [0.845, 1.000]	3.6×10^{-5}	1.3×10^{-3}
Rollout-Bellman	0.10	0.972 [0.873, 1.000]	1.3×10^{-5}	3.9×10^{-3}
Rollout-Bellman	0.05	0.973 [0.909, 1.000]	8.4×10^{-6}	1.9×10^{-3}
Rollout-Bellman	0.02	0.975 [0.933, 1.000]	8.4×10^{-6}	7.2×10^{-4}

C.6 EXACT-SAMPLER MATRIX PER SAMPLING SEED

Table 27 separates the two sampling seeds of Table 4 for Spec-4 against CARS and lists the recorded target forwards, target rows and draft forwards of each history, which are identical over the seven timing rotations of a seed. Spec-4 acquires about as many target rows as CARS but scores a drafted span in one forward, so it needs fewer target forwards in every long-prompt cell; with a backend that recomputes the full prefix at each forward, this is where its wall-clock advantage arises.

Table 27: Exact-sampler matrix by sampling seed, each entry giving seed 1 / seed 2. Ratio: Spec-4/CARS wall clock, geometric mean over the seven timing rotations of the seed. Forwards and rows: recorded target forwards and target rows per history, in the order C/S/E for CARS, Spec-4 and suffix-shared enumeration. Draft: Spec-4 draft forwards.

Support	Prompt	Outputs	Spec-4/CARS	Target forwards C/S/E	Target rows C/S/E	Draft
4	short	1	1.277 / 1.371	3/3/4 ; 5/5/4	6/7/10 ; 8/10/10	1 ; 2
4	short	16	1.021 / 1.021	7/5/4 ; 7/5/4	10/10/10 ; 10/10/10	2 ; 2
4	long	1	0.788 / 0.841	3/2/4 ; 4/3/4	6/6/10 ; 7/7/10	1 ; 1
4	long	16	0.814 / 0.783	7/5/4 ; 6/4/4	10/10/10 ; 9/9/10	2 ; 2
16	short	1	0.912 / 0.633	15/6/16 ; 12/3/16	18/16/34 ; 15/10/34	7 ; 4
16	short	16	1.192 / 1.041	20/10/16 ; 22/10/16	23/26/34 ; 25/25/34	13 ; 12
16	long	1	0.614 / 0.941	5/2/16 ; 5/3/16	8/8/34 ; 8/11/34	3 ; 5
16	long	16	0.657 / 0.718	22/10/16 ; 22/11/16	25/26/34 ; 25/28/34	13 ; 14
64	short	1	1.038 / 1.212	7/3/64 ; 12/5/64	10/10/130 ; 15/17/130	4 ; 9
64	short	16	0.835 / 1.213	49/14/64 ; 54/24/64	52/42/130 ; 57/66/130	25 ; 39
64	long	1	1.245 / 0.962	7/5/64 ; 7/4/64	10/19/130 ; 10/15/130	11 ; 8
64	long	16	0.655 / 1.158	36/14/64 ; 22/16/64	39/45/130 ; 25/47/130	28 ; 28
256	short	1	1.807 / 1.909	16/9/256 ; 9/5/256	19/31/514 ; 12/20/514	19 ; 12
256	short	16	1.315 / 1.909	55/21/256 ; 52/26/256	58/72/514 ; 55/98/514	48 ; 69
256	long	1	0.569 / 0.567	9/3/256 ; 9/3/256	12/12/514 ; 12/12/514	6 ; 6
256	long	16	0.698 / 0.677	16/6/256 ; 20/7/256	19/24/514 ; 23/29/514	15 ; 19

C.7 COST STRUCTURE OF THE EXACT-SAMPLER BACKEND

A least-squares fit to all 1,568 timed runs, with one intercept per event and the recorded target forwards, target input tokens, draft forwards and draft input tokens as regressors, gives $R^2 = 0.9998$ and prices a target forward at 0.067 s plus 0.93 ms per input token and a draft forward at 0.072 s plus 0.15 ms per input token; the fit was registered as a descriptive analysis before it was computed. Every forward of an event has the same padded length, namely the prompt of 17 or 353 tokens plus the longest event path, so most of the per-token term is the cost of recomputing the prompt. Table 28 applies the fitted prices to the recorded counters of each history: once as recorded, and once with the prompt's tokens paid only by the first target and the first draft forward of a history, a modelled prefix-cached interface in which every later forward still pays its padded continuation. The break-even column gives, for that interface, the ratio of a target forward's fixed cost to a draft forward's above which Spec-4's modelled cost is below that of CARS. Modelled ratios are not measurements; the measured ratios are in Tables 4 and 27. The last column divides the one-off qualification of the row backend by the per-output wall clock of the fastest exact arm at 16 outputs.

D IMPLEMENTATION AND REPRODUCIBILITY

What the supplement contains. The anonymous supplement holds the source of the finite-trie reference builder for exact backward Φ, $\mu^\star$ and μ^{proj}, the DFlash verifier loop with residual and bonus sampling and cache cropping, with runners for tries A–B and C–E, the exact-sampler matrix with its row backend, the test fixtures of the samplers, and the records and recomputations listed in Table 29; its README names the file behind each row. The DFlash loop imports the public DFlash model code, and the sampler runner reads the checkpoints; neither the model code nor the weights are redistributed. Every recomputation in Table 29 runs from the supplied records and code alone; it does not rerun a model.

Table 28: Cost structure of the exact-sampler backend, with fitted prices applied to the recorded counters and ranges over output counts and sampling seeds. Prompt share: share of the fitted target cost spent on prompt tokens. Spec-4/CARS: modelled cost ratio as recorded and with the prompt paid once per history. Break-even: target-to-draft fixed-cost ratio above which Spec-4 is modelled cheaper with the prompt paid once. Amortization: outputs whose sampling time equals the one-off qualification. *At 16 outputs; at one output Spec-4 and CARS use the same number of target forwards on this event.

Event	$\lvert\mathcal{L}\rvert$	Prompt share	Spec-4/CARS, modelled		Break-even	Amortization
			as recorded	prompt once		outputs
short	4	0.18	0.96–1.35	1.01–1.40	0.96*	258
long	4	0.82	0.77–0.83	1.07–1.10	1.31–1.68	201
short	16	0.18	0.53–1.05	0.59–1.15	0.36–1.23	381
long	16	0.82	0.59–0.91	1.06–1.39	1.07–2.81	297
short	64	0.17	0.71–1.05	0.79–1.15	0.61–1.21	829
long	64	0.81	0.63–1.21	1.15–1.84	1.21–5.85	961
short	256	0.17	1.10–1.65	1.22–1.82	1.30–2.94	4,314
long	256	0.81	0.54–0.67	1.03–1.25	1.00–1.47	9,593

Table 29: Claim-to-record map of the supplement. "Archived" marks values read from a shipped record rather than recomputed.

Result	Records	Recomputation
Fidelity bounds, headline cells and further seeds, Tables 2, 3 and 12	full terminal count vectors and same-precision reference laws for all 40 cells of seed 42 and the 30 cells of the three further fp32 seeds	every seed TV from its counts; bounds rounded up
Numerical-path family, Table 14	the 40 seed-42 count vectors and reference laws above, with the archived Monte Carlo p-values of the 19,999-draw null calibrations	TV of every cell from its counts; Holm adjustment of the archived p-values
Field–value events, Table 22 and last column of Table 3	the fp32 count vectors and reference laws above	every field–value marginal error; frequencies in the verifier arms; simultaneous field–value bounds; head/tail full-law bounds of Section A.1
Exact samplers, Tables 4 and 27	1,568 timed runs with outputs and recorded forward and row counts; design; per-event loading and qualification times	paired ratios per seed; resolved cells; per-seed counts
Spec-4 and CARS exactness, Section E.3	the unmodified sampler code; registration; enumeration outputs	all 837 enumerated cases, in minutes
Stopping certificate, Section C.5 and Table 26	registration, runner and per-instance readings of the 60 instances; 3,950 saved acquisition orders on A and B	all local certificate checks; the certified-level summary from the readings
Exact baselines and ASAp, Table 5; tokenization counts	baseline records and ASAp runs; the unedited GAD estimator file	ASAp columns archived; GAD-versus-reimplementation comparison of complete sequences and final laws rerun
Detailed experiments of Section C, Tables 19 to 21, 24 and 25 and multi-seed sweeps	the run records with per-run laws, counts or TVs	archived
Estimator hierarchy, Table 23 and Figure 1	toy LM, grammar and estimator code; the exact-law reanalysis record	all induced laws, fixed-table draws and certificates; visitation-weighted target rows; the figure's depth laws
Second verifier, Table 15	registration, verifier source, count vectors and same-dtype reference laws of all 24 cells, with the archived Monte Carlo p-values	every TV and bound from its counts; Holm adjustment of the archived p-values

Table 29, continued

Result	Records	Recomputation
Second target, Table 16	registration, event definitions, count vectors, reference laws and run statistics of the 8 registered cells, with the archived Monte Carlo p-values	every TV, bound and supported label; Holm adjustment over the 8 registered cells
Schema-stating extension, Table 17	registration, event and prompt definitions, count vectors, reference laws and run statistics of the six cells, with the archived Monte Carlo p-values	every TV, bound and supported label; Holm adjustment over the 6 registered cells
Schema-stating cells on Qwen3-14B, Table 18	registration, count vectors, reference laws and run statistics of both halves of the six cells, with the archived Monte Carlo p-values	the registered analysis: every TV, bound and supported label; Holm adjustment over the 6 registered cells
Prompt crossover, Table 7	per-instance gaps and Φ_{root} of the 25 events under the three prompts, with the archived registered paired differences	every median, range and count of the table
Gap, rarity and dispersion, Table 8	the six node tables of the prompt panel: every non-terminal prefix of the 150 instances with the log-probabilities of its legal children, prompts and token paths	every gap and Φ_{root} of Table 7, D, the identity and bounds of Proposition 5, the association and its interval
Exact cost and certification on the panel, Table 9	the six node tables above; the unchanged CARS sampler code	CARS target rows for ten sampling seeds per instance; mass-first certificate traces with the bound of Proposition 9 and the actual TV
Small-tree check of Proposition 9 in Section F.6	the generating script with a fixed seed	every tree, completion set and linear program
Field–value outcomes on the panel, Table 10	the six node tables above	every field–value error, flipped modal value and group summary
Cost structure, Table 28	the 1,568 timed runs with their recorded counters; the event prompts; one-off costs	fit, prompt share, modelled ratios, break-even and amortization
Registered tests, Table 30	registrations, per-instance readings and analysis code of each registered study	three-layout readouts; the others are archived readings with their analysis code

Execution contract. The draft's parallel marginals are masked at the already drafted trie prefix; those same normalized proposal probabilities enter acceptance and residual sampling. The target scores a full block, accepts left-to-right, stops verification at the first rejection, and otherwise takes a bonus token. Target and draft caches are cropped to the committed-prefix convention. The canonical arm substitutes the stored reference logits for each target distribution; the native arm uses the block-forward logits. The loop uses two finite-precision safeguards, a 10^{-12} floor on the proposal denominator and a fallback for a vanishing residual. Recorded kernel discrepancies quantify the visited computations.

Timing contract of the exact samplers. The per-history clock of Table 4 starts after both models are loaded and the row backend has passed its numerical qualification, with an empty probability-row cache, and stops after the last output; it includes row acquisition, sampling and trace writing. Model loading, once per run, takes 15–22 s for the target and 4.2–5.5 s for the draft, and the qualification, an offline check against recomputed rows, takes 6.6 s to 2,896 s per event; both are reported separately. The qualification evaluates the row of every non-terminal prefix of the event under each checked acquisition geometry, evaluating shared ancestors repeatedly, and passes none of these rows to the timed samplers, which start from an empty cache; a deployment that kept those rows could instead sample from the full table, so the timed comparison concerns events whose rows are not already held. The backend recomputes the full prefix at each forward without a KV cache and pads to a fixed length. Each combination of event, arm and output count has two sampling seeds; the seven arm-order rotations of a seed reproduce the same outputs and serve as timing repeats, so the two sources of variation are reported separately in Table 27, and a cell counts as resolved only when both seeds give a ratio on the same side of one. The short prompts of 17 tokens state the required string

length in one sentence; the long prompts of 353 tokens put 16 repetitions of a fixed two-sentence description of the field before that sentence. The qualification is paid once per event and backend, not per output.

Recomputation from the supplement. From the supplement alone, the certificate checker reproduces its 30,576 local checks with zero violations, the three-layout readouts are recomputed from their per-instance inputs, the exact-law reanalysis reproduces every entry of Table 23 and the laws of Figure 1, the second-target statistics of Table 16 and the registered analyses of Tables 17 and 18 are recomputed from the count vectors, the exhaustive enumeration of Section E.3 reruns on the shipped sampler code, and the prompt-panel gaps and root survival probabilities are rederived from the shipped node tables. The target-row tables of the 60 unseen instances and of the draft-lookahead studies are summarized by their per-instance readings. Model executions are not rerun.

E Persistent Prefix Queries and Exact Adaptive Sampling

E.1 Fixed-law cache and cost contract

Fix the model, prompt, grammar, numerical path and a nonnegative lookahead horizon k. Each query returns the same canonical next-token log-probability row. $\mathrm{P}'(k)$ samples y with weight $p(y \mid v)\Phi_k(vy)$, where Φ_k is survival for the next k steps and terminal leaves absorb. Computing this kernel at v requires row v and the nonterminal rows at distances one through k below it. The executable sampler calls a row-query oracle for exactly these rows, applies the same truncated recursion, and normalizes. A memoized row is reused without changing the horizon.

We distinguish three lifetimes. An uncached oracle pays for each query. A request-local cache clears between delivered outputs. A persistent cache retains rows across the m outputs sharing the contract above. Rows are not shared between prompts, schemas or numerical paths. The exact enumeration comparator acquires every nonterminal row once and subsequently draws from its terminal categorical law without another model call.

Proposition 10 (Persistent lookahead acquisition)**.** *Let $A_x(k)$ be the set of rows demanded while generating terminal string x, let ν_k be the fixed $\mathrm{P}'(k)$ law, and let*

$$r_u = \Pr_{X\sim\nu_k}\big[u \in A_X(k)\big].$$

For m independent requests,

$$\mathbb{E}Q_m = \sum_{u\in\mathcal{N}} \big[1-(1-r_u)^m\big] \le |\mathcal{N}|, \tag{2}$$

where $\mathcal{N}$ is the set of nonterminal trie nodes. The cache leaves ν_k unchanged. On a trie, r_u is the visitation probability of the ancestor of u at depth $\max\{\mathrm{depth}(u)-k, 0\}$.

Proof. Step 1: the law is unchanged. A cached query returns the identical row used by an uncached query, so every local normalized transition and hence the terminal law is unchanged.

Step 2: the expected count. A row is acquired if at least one request demands it; independence gives probability $1-(1-r_u)^m$. Summing these indicators proves equation 2. The realized union is also a subset of $\mathcal{N}$.

Step 3: the visitation form. Among the ancestors within distance k of u, the earliest one is visited whenever any later one is visited. Its visitation event therefore equals the event that u is demanded. □

For comparison, the two other lifetimes cost

$$\text{request-local cache:}\quad m\sum_u r_u,$$

$$\text{no cache:}\quad m\,\mathbb{E}_{X\sim\nu_k}\sum_{v\in\mathrm{path}(X)}\big|\mathrm{cone}_k(v)\big|,$$

with the expectation under the lookahead's own law ν_k. A row costs one full-prefix model evaluation on the canonical backend; the supplementary data also give the expected cumulative input-token volume by weighting each summand of equation 2.

Example and checks. For Schema B at $k = 5$, all 16 rows are demanded on one request. Ten requests cost 590 uncached row evaluations, 160 with a request-local cache, and 16 with a persistent cache, the size of the complete table. The implementation passed eight deterministic test groups, including exhaustive one- and two-request enumeration on 75 random small tries, zero requests, absorbing leaves and saturated horizons. On the 60 registered unseen instances, 7,461 forced terminal-path/cache-mode comparisons matched the uncached terminal probabilities exactly in the stored floating-point calculation.

E.2 CARS ON THE LEAF-STOPPING EVENT

Retained future mass. At an unpruned prefix v, let

$$a(v) = 1 - \sum_{y \in A(v)} p(y \mid v)$$

be its aggregate invalid-edge mass. All such edges go to one failure sink; a valid leaf stops immediately, with no extra EOS factor, so the individual token identities of invalid edges are unnecessary and the absolute rows suffice. Initially the retained future mass is $g(v) = 1$, including at leaves. For a set B of bulk-pruned prefixes,

$$g(v) = \begin{cases} 1, & v \notin B \text{ or } v \text{ is a leaf}, \\ \displaystyle\sum_{y \in A(v)} p(y \mid v)\, g(vy), & v \in B. \end{cases}$$

Proposal and bulk update. CARS proposes each legal edge with unnormalized weight $p(y \mid v)g(vy)$, and the failure sink with weight $a(v)$ only when $v \notin B$. The weights sum to $g(v)$: the visited set is ancestor-closed, so an unpruned node has no pruned descendant. After every attempt, including success, *every* visited proper prefix enters B and the masses are updated bottom-up. This is the published bulk update (Parys et al., 2026), adapted to the declared terminal event. Only queried rows enter the sampling state; exact references used for checking are unavailable to the sampler.

Exactness and attempt count. On any accepted path x, the proposal factors telescope to $p(x)/g(\mathrm{root})$. Normalizing over valid leaves gives $\mu^\star$, independently of B. Updating between failed attempts therefore also preserves the accepted law. Each failure discovers at least one new prunable row, so at most N failures occur over an arbitrarily long persistent-cache run; M outputs require at most $M + N$ attempts and at most N distinct row queries.

Why the bulk update matters. For one legal token of probability 0.01 and 99 invalid tokens of probability 0.01, bulk CARS needs 1.99 attempts in expectation for its first output: an initial success costs one attempt; a failure removes all invalid tokens and the second attempt succeeds. Individual-token exclusion instead takes 50.5 attempts in expectation. Both acquire only one persistent row. The test fixtures of the released sampler code check, in exact rational arithmetic, that every local accept/residual step they execute reproduces its target row.

E.3 SPEC-4: DRAFT SPAN PROPOSALS INSIDE CARS

State. Spec-4 is the CARS sampler of Section E.2 with one change: where CARS acquires one row and draws one token, Spec-4 may acquire the rows of a drafted span in one target forward and verify the span. Its state is the append-only row cache $\mathcal{R}$, whose rows are exact and never change, the retained masses g and the pruned set B. At the start of every attempt it takes a snapshot (g, B) that is frozen for the whole attempt: rows acquired during the attempt, including rows of drafted prefixes that are not committed, enter g and B only in the update after the attempt ends, which sets B to the set of all cached prefixes. For a cached prefix v the frozen kernel over the legal children and the failure sink $\perp$ is

$$\kappa(v)(y) = p(y \mid v)\, g(vy)/g(v),$$
$$\kappa(v)(\perp) = a(v)\, \mathbf{1}[v \notin B]/g(v).$$

The draft law q_v at an uncached prefix is the draft model's next-token law restricted to the legal children of v and renormalized; it gives $\perp$ zero mass. At a prefix with a single legal child the proposal

is that child and no draft forward is made. Nothing is rolled back after a rejection: rows acquired for discarded draft tokens stay in the cache, and the discarded tokens are dropped.

Algorithm 1 One output of Spec-4 with span length $\ell = 4$. To extend v by a symbol y is to fail the attempt if $y = \perp$ and otherwise to set $v \leftarrow vy$; every drawn symbol, including a bonus, extends v exactly once.

loop
  freeze (g, B); $v \leftarrow$ root after the common unary chain
  while v is not a leaf **do**
    if $v \in \mathcal{R}$ **then**
      draw $y \sim \kappa(v)$ and extend v by y {no draft}
    else if no child of v is an uncached nonterminal **then**
      acquire row v; draw $y \sim \kappa(v)$ and extend v by y {a CARS step}
    else
      draft $d_1, \ldots, d_j$ with $j \leq \ell$ from q, starting at v, stopping at a leaf or a cached prefix
      acquire the rows of the drafted prefixes, and of the prefix after d_j if it is an uncached nonterminal, in one target forward
      for $i = 1, \ldots, j$ **do**
        with probability $\min\{1, \kappa(v)(d_i)/q_v(d_i)\}$ set $y \leftarrow d_i$; otherwise draw $y \propto (\kappa(v) - q_v)^+$ and mark rejected
        extend v by y; **if** rejected or v is a leaf **then** leave the span
      end for
      if no rejection and v is not a leaf **then** draw $y \sim \kappa(v)$ and extend v by y {bonus}
    end if
  end while
  update g bottom-up over all cached prefixes and set $B \leftarrow \mathcal{R}$
  if the attempt did not fail **then return** v
end loop

Exactness. Proposition 11, proved in Section F.7, shows that every delivered output has law $\mu^\star$ given the entire past, for every reachable cache state; outputs that share a persistent cache are therefore independent draws from $\mu^\star$. The static-span arms of the released records, which propose spans at every node, are covered by the same argument.

Exhaustive trajectory enumeration. A registered check runs the unmodified sampler code of the recorded matrix and replaces only its random choice by an enumerator that follows every choice with positive probability, so every complete trajectory is visited with its exact rational probability. It covers 30 random trees with 2–5 leaves, depth at most 3 and the legal mass of each row drawn from $\{1, \frac{9}{10}, \frac{3}{5}, \frac{1}{5}\}$; three hand-built tries, namely a valid leaf of mass $\frac{1}{200}$ behind a failure sink of mass 0.985, a draft that puts 0.98 on the least likely child, and a unary chain above the first branching node; and six deep tries of depth 5–7, each with an independent random draft and an adversarial draft that puts 0.9 on the least likely legal child, for CARS, Spec-4 and static spans 1, 2 and 4, and for 1, 2 and 3 outputs on one persistent cache. In all 837 cases, with 130,762 trajectories and 186 Spec-4 cases, the joint law of the outputs equals $(\mu^\star)^{\otimes m}$ exactly and the probabilities sum to one. The enumerated Spec-4 trajectories include rejections at each of the four span positions, whole accepted spans with a bonus draw, failure-sink draws from single-row steps and from residuals, envelope updates between attempts, and outputs drawn from rows cached for an earlier output.

F Proofs

F.1 Proof of Corollary 2

Step 1: the per-step rule. Fix a committed prefix $x_{<t}$ and $A_t = A_{\mathcal{C}}(x_{<t})$. By B3 the decoder has exact access to A_t, and by B1 it consults the constraint only through A_t. By B2 it draws $d_t \sim \tilde{q}_t$ and

accepts with probability $\min\{1, \tilde{p}_t(d_t)/\tilde{q}_t(d_t)\}$, where

$$\tilde{q}_t(y) = \frac{q_t(y)\,\mathbf{1}[y \in A_t]}{Z_{q,t}},$$
$$\tilde{p}_t(y) = \frac{p_t(y)\,\mathbf{1}[y \in A_t]}{Z_{p,t}},$$

and otherwise it samples from $(\tilde{p}_t - \tilde{q}_t)^+/Z^+$.

Step 2: the committed token. The supplied-target identity in Appendix A.1 of Leviathan et al. (2023) states that the committed token is distributed as $\tilde{p}_t = \mu_t^{\text{proj}}$ for any proposal $\tilde{q}_t$; the residual step is what makes the marginal exact.

Step 3: the sequence law. The chain rule gives

$$\Pr(x_{1:T}) = \prod_t \mu_t^{\text{proj}}(x_t \mid x_{<t}) = \mu^{\text{proj}}(x_{1:T}).$$

Step 4: the block verifier. The block verifier of Section 5.1 uses fixed per-position raw draft logits and draws each proposal after masking to the already drafted grammar prefix. Its $q^{(p)}$ is that conditional masked proposal, not an unconditional marginal of a correlated block. The verifier scores position p against the target conditional given the accepted draft prefix, and stops at the first rejection; each verified position is therefore an instance of the rule of Step 1 with proposal $q^{(p)}$, so the corollary applies to it position by position. □

F.2 Proof of Theorem 6

Normalization lemma. For non-negative vectors a, b over a finite set with $A = \sum_y a_y > 0$ and $B = \sum_y b_y > 0$,

$$\mathrm{TV}\big(a/A,\, b/B\big) \le \frac{\sum_y |a_y - b_y|}{A}.$$

Proof. Let $d = \sum_y |a_y - b_y|$, so that $d \ge |B - A|$. Then

$$\begin{aligned}\frac{1}{2}\sum_y \Big|\frac{a_y}{A} - \frac{b_y}{B}\Big| &\le \frac{1}{2}\sum_y \frac{|a_y - b_y|}{A} + \frac{1}{2}\sum_y b_y \Big|\frac{1}{A} - \frac{1}{B}\Big| \\ &= \frac{d}{2A} + \frac{|B - A|}{2A} \\ &\le \frac{d}{A}. \qquad \square\end{aligned}$$

(a) Additive. Take $a_y = p_t(y)\Phi_t(y)$ and $b_y = p_t(y)\hat{\Phi}_t(y)$ on A_t. Then

$$A = Z_p\,\bar{\Phi}_t,$$
$$d = \sum_y p_t(y)\,\big|\hat{\Phi}_t(y) - \Phi_t(y)\big| = Z_p\,\bar{\delta}_t,$$

so the lemma gives

$$\begin{aligned}\mathrm{TV}(\hat{\mu}_t, \mu_t^\star) &\le \frac{\bar{\delta}_t}{\bar{\Phi}_t} \\ &\le \frac{\delta_t}{\bar{\Phi}_t}.\end{aligned}$$

(b) Relative. With $b_y = r_y a_y$,

$$\begin{aligned} d &= \sum_y a_y \, |r_y - 1| \\ &= A \, \mathbb{E}_{\mu_t^\star} |r_Y - 1|, \end{aligned}$$

so

$$\begin{aligned} \mathrm{TV}(\hat{\mu}_t, \mu_t^\star) &\le \mathbb{E}_{\mu_t^\star} |r_Y - 1| \\ &\le \max_y |r_y - 1|. \end{aligned}$$

Tokens with $\Phi_t(y) = 0$ have $\hat{\Phi}_t(y) = 0$ under the multiplicative assumption and carry no mass under either law.

(c) Sequence. For laws P, Q over sequences with kernels P_t, Q_t, the standard coupling argument gives

$$\mathrm{TV}(P, Q) \le \sum_t \mathbb{E}_{x_{<t} \sim P} \Big[\mathrm{TV}\big(P_t(\cdot \mid x_{<t}),\, Q_t(\cdot \mid x_{<t})\big) \Big];$$

apply (a) or (b) prefix by prefix, and use $\mathrm{TV} \le 1$ inside the expectation. Summing the exact per-prefix TV instead of its bound gives the quantity we call the *sequence sum* in Section C.4; both dominate the terminal-law TV. □

Comparison with denominator-based bounds. The additive bound $\delta_t / (\bar{\Phi}_t - \delta_t)$ follows from (a) by the cruder estimate $\hat{Z} \ge Z_p(\bar{\Phi}_t - \delta_t)$ for the denominator; the relative bound $\epsilon/(1-\epsilon)$ likewise. Both are implied by, and never tighter than, (a) and (b).

F.3 Proof of Proposition 7

Step 1: the source problem. We reduce from #NFA: given a nondeterministic finite automaton M over Σ and a length n written in unary, count the strings of length exactly n accepted by M; this problem is #P-hard (Álvarez & Jenner, 1993). An NFA converts in polynomial time into a right-linear grammar G with $\mathcal{L}(G) = \mathcal{L}(M)$, a special case of a CFG, so it suffices to count $|\mathcal{L}(G) \cap \Sigma^n|$.

Step 2: the oracle answers the count. Set p uniform over Σ, horizon n, and the empty prefix. For $y \in \Sigma$,

$$\Phi_1(y) = \frac{\big|\{w \in \mathcal{L}(G) \cap \Sigma^n : \, w_1 = y\}\big|}{|\Sigma|^{n-1}},$$

so that

$$\sum_y \Phi_1(y) \, |\Sigma|^{n-1} = \big|\mathcal{L}(G) \cap \Sigma^n\big|.$$

Hence an oracle for Φ_t solves the counting problem with $|\Sigma|$ queries, a polynomial-time Turing reduction; the unary encoding of n keeps the horizon polynomial in the input size.

Step 3: the finite-language cost. The cost statement is immediate from the backward recursion over the trie,

$$\Phi(s) = \sum_{y \in A(s)} p(y \mid s) \, \Phi(sy),$$

which reads $p(\cdot \mid s)$ once per non-terminal node. □

F.4 Proof of equation 1

By definition and by the backward recursion,

$$\begin{aligned} \hat{\Phi}_t^{\text{ls-true}}(y) &= \sum_{u \in A_{t+1}(x_{<t} y)} p(u \mid x_{<t} y), \\ \Phi_t(y) &= \sum_{u \in A_{t+1}(x_{<t} y)} p(u \mid x_{<t} y) \, \Phi_{t+1}(u \mid x_{<t} y). \end{aligned}$$

Subtracting gives equation 1; it is non-negative because $\Phi_{t+1} \le 1$, and at most $\hat{\Phi}_t^{\text{1s-true}}(y)$ when every successor is a dead end. □

F.5 KL IDENTITY AND PATH-MASS DISPERSION

For $y \in A_t$,

$$\mu_t^\star(y) = p_t(y)\Phi_t(y)/Z^\star,$$
$$\mu_t^{\text{proj}}(y) = p_t(y)/Z_p,$$

so, with $\bar{\Phi}_t = Z^\star/Z_p$,

$$D_{\mathrm{KL}}(\mu_t^\star \| \mu_t^{\text{proj}}) = \sum_y \mu_t^\star(y) \log \frac{\Phi_t(y) Z_p}{Z^\star}$$
$$= \mathbb{E}_{\mu_t^\star}\big[\log\big(\Phi_t(Y)/\bar{\Phi}_t\big)\big]. \qquad \square$$

Proof of Proposition 5. Every prefix reached with positive probability has $Z_p(x_{<t}) > 0$ because the constraint has no dead ends. For a complete path $x \in \mathcal{L}(\mathcal{C})$,

$$\mu^{\text{proj}}(x) = \prod_{t \le |x|} \frac{p_t(x_t)}{Z_p(x_{<t})} = \frac{p(x)}{W(x)},$$
$$\mu^\star(x) = \frac{p(x)}{Z_\mathcal{C}} = \mu^{\text{proj}}(x)\,\frac{W(x)}{Z_\mathcal{C}},$$

and summing $\mu^{\text{proj}}(x)W(x) = p(x)$ over $\mathcal{L}(\mathcal{C})$ gives $\mathbb{E}_{\mu^{\text{proj}}}[W] = Z_\mathcal{C}$. Hence

$$\mathrm{TV}(\mu^{\text{proj}}, \mu^\star) = \tfrac{1}{2}\sum_x \mu^{\text{proj}}(x)\left|\frac{W(x)}{Z_\mathcal{C}} - 1\right| = \frac{\mathbb{E}_{\mu^{\text{proj}}}\big[(W - Z_\mathcal{C})^+\big]}{Z_\mathcal{C}},$$

where the second form uses $\mathbb{E}_{\mu^{\text{proj}}}[W/Z_\mathcal{C} - 1] = 0$. Since $0 \le W \le 1$ and $w \mapsto (w-m)^+$ is convex with value 0 at $w = 0$ and $1 - m$ at $w = 1$, it lies below the chord: $(w-m)^+ \le w(1-m)$ on $[0,1]$. With $m = Z_\mathcal{C}$,

$$\mathbb{E}_{\mu^{\text{proj}}}\big[(W - Z_\mathcal{C})^+\big] \le (1 - Z_\mathcal{C})\,\mathbb{E}_{\mu^{\text{proj}}}[W] = Z_\mathcal{C}(1 - Z_\mathcal{C}),$$

so $\mathrm{TV}(\mu^{\text{proj}}, \mu^\star) \le 1 - Z_\mathcal{C}$. The KL statement is the chain rule for the two sequence laws, whose per-step kernels at every prefix are $\mu_t^\star$ and μ_t^{proj}. □

F.6 OBSERVABLE CERTIFICATES

Support assumptions. The finite tree includes the complete conditioning event, including its termination rule; in our replay it is the canonical compact-JSON tokenization with leaf stopping. Prefix probabilities are those of one fixed numerical reference. Queries reveal all legal children at a prefix; unrevealed probabilities, exact future masses and terminal truth are inaccessible to query selection. The topology is supplied, so its construction is part of the decoder's setup.

Proof of Proposition 8. Step 1: containment. A valid leaf has survival probability one, a dead end zero, and an unresolved nonterminal has probability in $[0,1]$. If every child interval contains its survival probability, nonnegativity of the revealed p_y implies

$$\sum_y p_y L_y \;\le\; \sum_y p_y \Phi_y \;\le\; \sum_y p_y U_y.$$

Backward induction proves containment. Revealing an unresolved node replaces $[0,1]$ by a contained interval, and the same induction applies after every query, regardless of the rule choosing that query. This is deterministic containment, not a confidence statement requiring a union bound.

Step 2: the per-step bound. Write

$$a_y = p_y \Phi_y, \qquad c_y = p_y m_y,$$
$$A = \textstyle\sum_y a_y > 0, \qquad M = \textstyle\sum_y c_y > 0.$$

Then

$$d := \sum_y |a_y - c_y| \le \frac{1}{2} \sum_y p_y(U_y - L_y),$$

and the normalization lemma of Appendix F.2, with c as its first argument, gives

$$\mathrm{TV}(a/A,\, c/M) \le \frac{d}{M} \le b_t.$$

Since $U_y - L_y \le U_y + L_y$, $b_t \le 1$.

Step 3: the sequence claim. For completeness, compare two length-T laws by hybrids whose first t transitions use q and remaining transitions use $\mu^\star$. Consecutive hybrids share the prefix law q; contraction of TV by the common suffix kernel bounds their difference by

$$\mathbb{E}_{S_t \sim q}\, \mathrm{TV}\big(q_t(\cdot \mid S_t),\, \mu_t^\star(\cdot \mid S_t)\big).$$

The triangle inequality over t proves the sequence claim. Absorbing terminal states make variable-length sequences fit this argument. At a prefix with zero target survival, the conditional reference kernel is undefined: choose an arbitrary extension there and charge one. Such a prefix cannot occur under the true conditional law. An online adaptive decoder requires its cache/history as part of the state; one must use the corresponding conditional K_t and conditional certificate, not an unconditional root-only number.

Step 4: the numerical decomposition. Applying the same hybrid argument and the per-prefix triangle inequality

$$\mathrm{TV}(K_t, \mu_t^\star) \le e_t + b_t$$

proves the numerical decomposition of Section 4.1. □

A terminal bound when unresolved branches are omitted. Let a connected queried tree contain resolved terminal mass $L > 0$ and unresolved frontier upper mass $R = \sum_{s \in F} p(s)$, with root-to-frontier probabilities fully revealed. The distribution q_{leaf} obtained by normalizing the resolved terminal masses satisfies

$$\mathrm{TV}(q_{\mathrm{leaf}}, \mu^\star) \le \frac{R}{L+R}.$$

Indeed the unresolved valid mass is some $M \in [0, R]$; the true law mixes q_{leaf} with a distribution on disjoint unresolved leaves in proportions $L/(L+M)$ and $M/(L+M)$. Its TV from q_{leaf} is exactly $M/(L+M)$. This gives a truncation stopping rule that is informative when R is small; omitted valid leaves have zero output probability and their total conditional mass is explicitly bounded. For any complete prior-based estimate $\widehat{q}$, the triangle inequality gives

$$\mathrm{TV}(\widehat{q}, \mu^\star) \le \min\Big\{1,\ \mathrm{TV}(\widehat{q}, q_{\mathrm{leaf}}) + \frac{R}{L+R}\Big\}.$$

If $L = 0$, only the trivial bound one is used. The released checker evaluates this certificate for every saved query order without consulting hidden probabilities in the certificate computation (Section C.5).

Proof of Proposition 9. Let F be the unqueried frontier, $c = L/(L+R)$, so that $1 - c = R/(L+R)$, and let q be any law on valid leaves computed from the acquired rows and the topology; put $a = q(\text{resolved leaves})$. The rows at and below each $f \in F$ are unconstrained by the acquired information, and each of these prefixes has a next token outside the event. Sending the mass of the row at f to such a token makes the valid mass below f arbitrarily small; sending it along the path to one leaf makes that leaf's mass arbitrarily close to $p(f)$. Every bound below is therefore a limit of attainable values. Without an out-of-event token the valid mass below f can be pinned, for instance at $p(f)$ when every continuation of f is valid, and the bound need not hold.

First case: vanishing unresolved mass. Let the valid mass below every $f \in F$ tend to zero. Since $L > 0$, $\mu^\star(\text{unresolved leaves}) \to 0$, and since $\mathrm{TV}(q, \mu) \ge q(S) - \mu(S)$ for every set S,

$$\sup \mathrm{TV}(q, \mu^\star) \ge 1 - a.$$

Second case: full unresolved mass on the least-covered leaves. Give every $f \in F$ its full mass $p(f)$, concentrated on the leaf ℓ_f below f with the smallest q-probability. The limit law μ_1 has

$$\mu_1(x) = \frac{p(x)}{L+R} \text{ on resolved leaves}, \qquad \mu_1(\ell_f) = \frac{p(f)}{L+R},$$

and zero elsewhere. Each f has at least m valid leaves, so $q(\ell_f) \le q(\text{leaves below } f)/m$ and $\sum_{f\in F} q(\ell_f) \le (1-a)/m$. With $\mathrm{TV}(q,\mu_1) = 1 - \sum_x \min\{q(x),\mu_1(x)\}$ and $\sum_{\text{resolved}} \min\{q,\mu_1\} \le \min\{a,c\}$,

$$\mathrm{TV}(q,\mu_1) \ge 1 - \min\{a,c\} - \frac{1-a}{m}.$$

Combining. The supremum is at least

$$g(a) = \max\Big\{1-a,\ 1-\min\{a,c\} - \frac{1-a}{m}\Big\}.$$

If $a \le c$, then $g(a) \ge 1-a \ge 1-c$. If $a > c$, write $u = 1-a$; then $g = \max\{u,\ (1-c) - u/m\}$, whose minimum over u is attained where the two terms meet, at $u = (1-c)\,m/(m+1)$. Hence

$$\inf_a g(a) \ge \frac{m}{m+1}\,(1-c) = \frac{m}{m+1}\,\frac{R}{L+R}.$$

For q_{leaf} the terminal bound above is attained as the unresolved valid mass tends to R, which proves the second claim. □

The assumptions are needed, and the bound is attained. A linear program checks Proposition 9 on 300 random finite tries of depth 2–3 with one to three children per prefix, every leaf valid, a random connected set of queried prefixes containing the root, and acquired rows with an out-of-event remainder. The adversary chooses, for every unqueried frontier prefix f, zero valid mass below f or its full prefix mass $p(f)$ on one leaf below f; these completions are realizable under the proposition's assumptions and form a subset of all completions, so the program's value $V = \min_q \max \mathrm{TV}(q,\mu)$ is a lower bound on the minimax TV. In all 300 instances $V \ge \frac{m}{m+1}\frac{R}{L+R}$, with equality in 170, and $V \le 0.93\,R/(L+R)$ with median $0.68\,R/(L+R)$. When instead every continuation below the frontier is valid, so that the valid mass below f is pinned at $p(f)$, the value falls below the bound in 266 of the 300 instances and stays at most $0.89\,R/(L+R)$. Over the vocabulary $\{0,1\}$ with accepted leaves $\{0,10,11\}$, root row $(1/2,1/2)$ and prefix 1 unqueried, the pinned value is $1/4$ against the bound $1/3$; with an out-of-event token at prefix 1 it is $1/3$.

F.7 Exactness of Spec-4

Algorithm 1 starts below the common unary chain u, the longest prefix whose proper prefixes each have a single legal child, so every valid leaf is $x = uz$. Throughout this subsection u is absorbed into the prompt: p denotes $p_u(z) = p(uz)/p(u)$, the root is u, and ancestor closure, g and the products below refer to the subtree under u. Since $p(u)$ is common to all valid leaves, $\mu^\star(uz) \propto p_u(z)$, and without a common chain u is the empty prefix and $p_u = p$. For example, with valid leaves 00 and 01, first-token probability $1/2$ and second-token probabilities $2/5$ and $3/5$, the root is $u = 0$ with $g(u) = 1$, and an attempt returns 00 with probability $p_u(0) = 2/5 = \mu^\star(00)$, whereas the untransformed $p(00)$ is $1/5$.

Proposition 11 (Spec-4 is exact for every reachable state)**.** *Let $\mathcal{F}_k$ collect all randomness before attempt k of Algorithm 1, and let $(\mathcal{R}_k, g_k, B_k)$ be the state frozen at its start. For every valid leaf x,*

$$\Pr[\textit{attempt } k \textit{ returns } x \mid \mathcal{F}_k] = p(x)/g_k(\mathrm{root}),$$

with p and the root taken below the common unary chain. Consequently the output delivered by a request has law $\mu^\star$ given everything before the request, and the outputs of successive requests on one persistent cache are independent draws from $\mu^\star$. At most $|\mathcal{N}|$ attempts fail over any run, with $\mathcal{N}$ the nonterminal prefixes.

Proof. Step 1: the frozen kernel is a distribution. Every acquired row belongs to a committed prefix or to a drafted chain that starts at a committed prefix, whose ancestors are all cached; hence $\mathcal{R}_k$ is ancestor-closed. The update sets $B_k = \mathcal{R}_k$ and $g_k(v) = \sum_{y\in A(v)} p(y \mid v) g_k(vy)$ on B_k. If $v \notin B_k$, no descendant of v is cached, so $g_k(vy) = 1$ for every child and $\sum_y p(y \mid v) g_k(vy) + a(v) = 1 = g_k(v)$. In both cases $\kappa(v)$ sums to one. The implementation checks this identity at every kernel evaluation.

Step 2: one committed symbol. Fix the committed prefix v and the frozen state. A direct step and a bonus step draw from $\kappa(v)$, and each drawn symbol is committed once. In a span the draft token at v is drawn from q_v, which depends on v only, and is accepted with probability $\min\{1, \kappa(v)(d)/q_v(d)\}$; otherwise a symbol is drawn from the normalized residual. The committed symbol y then has probability

$$\min\{q_v(y), \kappa(v)(y)\} + \big(\kappa(v)(y) - q_v(y)\big)^+ = \kappa(v)(y),$$

including $y = \perp$, which has $q_v(\perp) = 0$ and is reached only through the residual. The next draft token was drawn from q_{vy} before verification, so given acceptance at v the same argument applies at vy; after a rejection the span ends and the remaining draft tokens are discarded. The kernel uses the frozen (g_k, B_k) and the exact row of v, which is acquired before v is verified; rows acquired later in the attempt do not enter it. Hence, given $\mathcal{F}_k$ and the committed prefix, every committed symbol has law $\kappa(v)$, whichever of the three step types produced it.

Step 3: the attempt law. Along a valid leaf $x = x_1 \cdots x_n$ the product of the kernels telescopes:

$$\prod_{t=1}^{n} \frac{p(x_t \mid x_{<t})\, g_k(x_{\le t})}{g_k(x_{<t})} = \frac{p(x)\, g_k(x)}{g_k(\mathrm{root})} = \frac{p(x)}{g_k(\mathrm{root})},$$

since $g_k = 1$ at leaves. This proves the displayed identity.

Step 4: the delivered output. The request returns the first successful attempt. Summing over k,

$$\Pr[\text{output} = x \mid \mathcal{F}_1] = p(x)\, \mathbb{E}\Big[\sum_k \mathbf{1}[\text{attempts} < k \text{ fail}]\, g_k(\mathrm{root})^{-1} \;\Big|\; \mathcal{F}_1\Big],$$

and the expectation does not depend on x, so the output has law $\mu^\star$ given $\mathcal{F}_1$, which contains all earlier outputs.

Step 5: failures. $\kappa(v)(\perp) > 0$ only if $v \notin B_k$. A failure at such a v acquires its row, so $v \in B_{k+1}$ and $\perp$ can no longer be drawn at v. Each failure therefore removes a distinct nonterminal prefix from the set at which $\perp$ has positive mass. □

G Registered Tests

Several studies were registered before their first model call, with decision rules fixed in advance. Table 30 lists every primary decision rule of these studies with its readout and registered label; secondary endpoints and subsidiary decisions are in the registrations and readings of the supplement. The last row is different in kind: a secondary analysis of the prompt-panel readings, registered before it was computed but after the per-instance gaps and root masses it uses had been recorded.

Table 30: Registered rules and readouts. Instances are schema–prompt pairs; "anticipated" means $\Phi_{\mathrm{root}} \ge 10^{-2}$; intervals are schema-bootstrap 95% intervals.

Study	Registered rule	Readout
Unseen schemas: masking gap	schema-level median, its 95% lower bound, and at least half of the 60 instances above 0.02 TV	met: median 0.465 [0.388, 0.756]; 60/60 above 0.02
Unseen schemas: lookahead cost	a k with TV ≤ 0.02 in $\ge 80\%$ of instances; at $m = 10$, schema-median saving over the cheapest exact comparator $\ge 20\%$ with 95% lower bound $\ge 5\%$	k exists in 60/60, $k =$ 1–17; saving: not met; registered score 4.9–69$\times$ the comparator
Certified stopping	mass-first bound ≤ 0.02 within 75% of full-table queries in $\ge 80\%$ of instances; no step with actual TV above bound by $> 10^{-10}$	18/60 instances; median query fraction 0.934; 0 violations
Further fp32 seeds	corrected cells not Holm-rejected and inside pointwise 99% envelopes	0/15 rejected; 14/15 inside; D, seed 2: 0.0166 vs. 0.0165; label: mixed
Exact-sampler matrix, 8 events, 1 and 16 outputs	primary comparison, registered without a threshold: Spec-4 vs. enumeration wall clock, each event and output count separately	Table 4, Spec-4/Enum. columns, 0.019–1.437

Table 30, continued

Study	Registered rule	Readout
Second verifier with chain draft, P1	the 4 fp32 native masked cells not Holm-rejected against μ^{proj}, masked TV within the null 95% quantile of the gap	0/4 rejected; 4/4 within; label: supported
Second verifier, P2	the 4 fp32 native corrected cells not Holm-rejected against $\mu^{\star}$, TV far below the gap; simultaneous 95% bounds	0/4 rejected; bounds 0.019–0.041 against gaps 0.19–0.62; label: supported
Second verifier, P3	native-vs-canonical row difference larger in bf16 than in fp32, aggregation not specified and read as the family maximum; Holm rejections among the 8 bf16 corrected cells only in native cells	family maximum 2.8×10^{-2} vs. 1.6×10^{-5}, per try larger in bf16 on A, C, D and below 5×10^{-6} in both on B; 1 rejection, a native cell on A; label: supported for the family maximum
Second target Qwen3-14B and cross-field events F–I, P1	the 8 masked cells not Holm-rejected against μ^{proj}, reported	8 of 8 cells reported; 0/8 rejected
Second target, P2	a corrected cell is supported if not Holm-rejected against $\mu^{\star}$ with Holm over 8 and its 95% bound is below half its gap; cells with gap < 0.05 not counted	0/8 rejected; supported: Qwen3-14B 3 of 3 counted, Qwen3-8B 3 of 3 counted; F not counted, gaps 0.021 and 0.032; final
Schema-stating prompts on Qwen3-8B, six cells with recorded gap ≥ 0.10, P1	the 6 masked cells not Holm-rejected against μ^{proj}, reported	6 of 6 cells reported; 0/6 rejected; final
Schema-stating prompts, P2	rule of the second-target P2, Holm over 6	0/6 rejected; supported: P_S 3 of 4 counted, P_N 2 of 2 counted; bounds 0.027–0.084, gaps 0.107–0.406; final
Schema-stating prompts on the second target Qwen3-14B, the six cells above, each run as two halves, P1	the 6 masked cells not Holm-rejected against μ^{proj}, reported	6 of 6 cells complete in both halves; 0/6 rejected; final
Schema-stating prompts on Qwen3-14B, P2	rule of the second-target P2, Holm over 6	0/6 rejected; Qwen3-14B gaps 0.014–0.150; supported: P_S 1 of 3 counted, U09 with bound 0.056 against gap 0.150; U06 and U08 with bounds 0.075 and 0.083 against gaps 0.087 and 0.068; not counted: U19 under P_S and both P_N cells, gaps 0.048, 0.032 and 0.014; final
Spec-4 and CARS exactness, exhaustive enumeration	the enumerated joint output law equals $(\mu^{\star})^{\otimes m}$ exactly in every case	837/837 cases exact
Kernel replay of the four rejected bf16 native cells	a cell is explained if EF ≥ 0.80 and replay $p \geq 0.01$; the family is explained if $\geq 80\%$ of cells are, partially if 50–80%	3 of 4 explained; label: partial
Draft-block lookahead, Qwen3-8B with DFlash-b16	gap halved in $\geq 80\%$ of the 60 instances: pass; 50–80%: partial; $< 50\%$: null	17/60; label: null
Chain-draft lookahead, Qwen3.5-9B with Qwen3.5-0.8B	same rule	22/60; label: null
Schema-stating raw prompts, 25 tries, 2 prompts per model	small gap: $\geq 80\%$ anticipated, median ≤ 0.02, upper bound ≤ 0.05, and $\geq 80\%$ of anticipated gaps ≤ 0.02; large gap: $\geq 80\%$ anticipated, median ≥ 0.10, lower bound > 0.05; out of regime: $< 80\%$ anticipated; else inconclusive. Cross-model link: agree / disagree / undetermined	Qwen3-8B: inconclusive (48/50; 0.037 [0.029, 0.050]); Qwen3.5-9B: inconclusive (49/50; 0.019 [0.010, 0.027]); link: undetermined; secondary, paired reduction from P_J to P_S and P_N: Qwen3-8B 0.358, 0.464; Qwen3.5-9B 0.234, 0.272; Table 7

Table 30, continued

Study	Registered rule	Readout
Schema-stating prompt, 98 enumerable schemas of JSONSchemaBench (Geng et al., 2025)	pass: median gap > 0.05 and lower bound > 0.02; null: upper bound < 0.02; fewer than 20 anticipated schemas: underpowered, report n and interval only; link arm on Qwen3-8B	11 anticipated on Qwen3.5-9B: underpowered, $n = 11$, interval [0.001, 0.488]; link with $n = 1$: undetermined
Three-layout event, 75 schema-stating instances per model: V1 gap	rules of the raw-prompt test, gap over parsed values	Qwen3.5-9B: inconclusive (74/75; 0.066 [0.031, 0.098]); Qwen3-8B: inconclusive (70/75; 0.067 [0.050, 0.078]); Ministral-3-8B: out of regime (51/75; 0.039 [0.030, 0.057])
Three-layout event: V2 coverage	median share of the three layouts among schema-valid samples: pass if ≥ 0.9 with lower bound ≥ 0.8; fail if < 0.5; else partial	Qwen3.5-9B: fail (0.291 [0.253, 0.384]); Qwen3-8B: fail (0.231 [0.188, 0.274]); Ministral-3-8B: partial (0.599 [0.474, 0.660])
Three-layout event: V3 tokenization	share of non-canonical tokenizations among sampled event hits: negligible if its 95% upper bound is ≤ 0.05, else material	Qwen3.5-9B: negligible (123/11,889; bound 0.012); Qwen3-8B: material (34,949/50,947; 0.690); Ministral-3-8B: material (8,177/36,138; 0.231)
Three-layout event: V4 cross-model	agree / disagree / undetermined over the V1 labels	undetermined for all three model pairs
Gap and rarity on the 25-event prompt panel, 150 instances recomputed from node tables, secondary analysis of recorded readings	supported if the event-bootstrap 95% interval of Spearman $\rho(\text{gap}, -\log \Phi_{\text{root}})$ lies above 0; within-group ρ reported descriptively	$\rho = 0.66$ [0.59, 0.72] with average ranks for ties: supported; within groups P_J, P_S, P_N: Qwen3-8B -0.48, 0.49, 0.44; Qwen3.5-9B -0.34, 0.25, 0.59