THE PREFIX IS ALREADY CACHED: REUSING TARGET KV FOR SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

A compact drafter can make several proposals before verifying against a single target model. Head-based speculative drafters use the target hidden states to build their proposals, to reconstruct a verified-prefix representation. Target model inference already cached the prefix representation in its own native key-value (KV) states. In same-query experiments, we show that a system with duplicate hidden KV states can change how future tokens are interacting with earlier context, and more importantly, how these states can cause inaccuracies in verified prefix readouts when queried by the target itself. Particularly we find that when a continuation visits information receiving low attention at the current token location, readouts can be subject to larger inaccuracies in these hidden KV state reconstructions, causing non-negligible errors in rejecting later steps. Based on these observations, we introduce OPSPEC, which removes the need for prefix reconstruction, where each branch simply queries the target's native KV states without any sequence linear drafter state over the verified token sequence. Operator Readout Distillation is used to distill the target model's prefix readouts with queries over these KV states, and a fused attention kernel allows for caching pages without copying. OPSPEC extends accepted length at 7 benchmarks on Qwen3-8B at the reported tree budget. Much larger gains are observed on tasks that require a larger re-visiting of earlier context. At 32K context length, OPSPEC saves 96.5% of incremental drafter state, allowing 20% more maximum batch size, and 9.7% more throughput at each method's maximum feasible batch.

1 INTRODUCTION

Speculative decoding replaces several sequential target-model steps with one parallel verification pass (Leviathan et al., 2023; Chen et al., 2023). A lightweight drafter proposes a continuation, and the target commits the longest prefix that satisfies its own acceptance rule. Practical speedup depends on the length of that committed prefix and the cost of producing and verifying the proposal. Recent methods improve this trade-off with target-conditioned heads, parallel token prediction, and tree-structured verification (Cai et al., 2024; Li et al., 2024b; 2025; Miao et al., 2024; Chen et al., 2024; 2026; Hu et al., 2026).

Treatment of the verified prefix has changed far less. Target inference already computes native key-value (KV) tensors for every verified token and keeps those tensors for subsequent decoding. Typical head-based drafters nonetheless retain several target hidden sequences, fuse them, and project a second contextual KV cache. Future candidates read this learned reconstruction. Whether it supports the same query-dependent computations as target KV has remained untested.

Cached KV is evaluated through a query, so two continuations can induce different computations over the same verified text. Consider a mathematics solution that states a boundary condition near the beginning and derives an intermediate value near the end. One candidate may continue from the recent derivation. Another may need to recover the earlier constraint. Their queries place weight on different prefix positions. A learned reconstruction can fit common next-token reads and still distort a less frequent read triggered by another candidate history.

We isolate this distortion by holding the future query fixed. Target queries are applied to native KV and to KV reconstructed by a normalization-matched causal-parallel drafter. Native KV reproduces the target prefix readout to numerical precision. Hidden-derived KV incurs 0.186 normalized

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.

error. Matching the target dimensionality reduces the error to 0.143, which shows that tensor shape alone does not recover the target computation. This same-query experiment locates an error in prefix reconstruction before draft-state prediction, candidate selection, or tree verification enters the comparison.

Finite query coverage explains why token supervision leaves the error unresolved. Training constrains a reconstructed cache through the queries observed on training trajectories. Many KV representations can agree on those readouts and disagree elsewhere. Candidate branches reveal the ambiguity when their histories redirect attention toward prefix evidence that was rarely queried during training. Reconstruction error and cross-branch variation in prefix attention both predict rejection after controlling for target entropy, token likelihood, branch rank, and depth. Including these quantities reduces the residual depth effect by 38.9%. Later positions are more vulnerable partly because they encounter more opportunities to require a new read of the prefix.

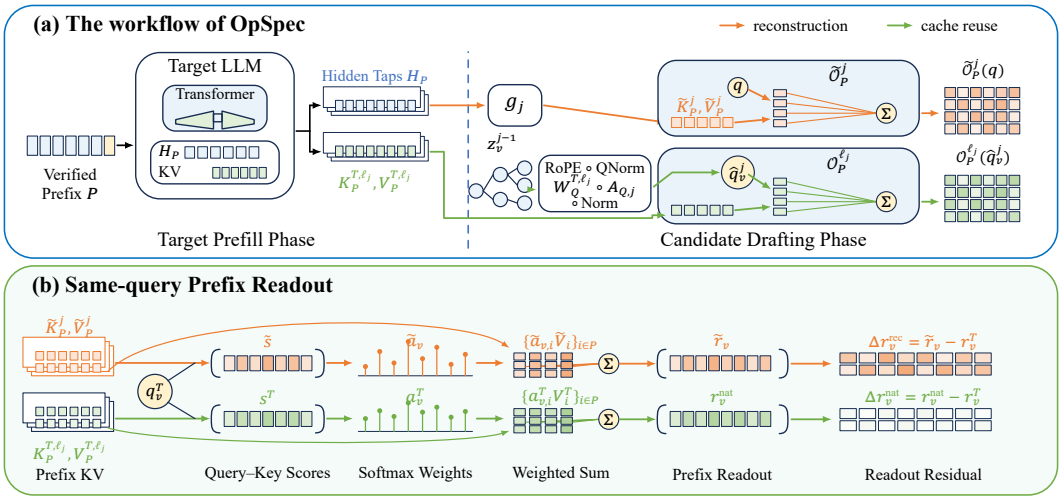


Figure 1: Reusing target-native prefix KV. (a) Hidden-conditioned drafting reconstructs a length-$|P|$ prefix state. OPSPEC aliases the target cache and retains only an anchor and $O(B)$ active-tree state. (b) Under the same target query, reconstructed and native KV yield NRMSE 0.186 and 0.0002.

OPSPEC removes prefix reconstruction from the drafting path. Its pretrained causal-parallel backbone still forms one state for each candidate branch. Full-prefix reconstruction is replaced by read-only access to selected target KV pages. A small adapter maps each branch state into the corresponding target attention space, allowing the resulting query to read native prefix KV. Speculative KV covers only the candidate and its ancestors. Shared-prefix attention and branch-local attention run in dedicated kernels and are merged with the exact softmax normalizer, so direct cache reuse does not materialize another prefix state.

Native KV fixes the prefix computation, leaving the quality of the branch query as the remaining readout error. Operator Readout Distillation supplies direct supervision for this query. A frozen target query is restricted to the verified prefix to produce the desired readout and attention normalizer. Draft queries learn these intermediate quantities together with the target token distribution. Candidate branches generated by the current drafter provide the final stage of training.

We test each link between reconstruction error and decoding performance. Same-query tests measure the prefix representation independently of draft-state quality. Controlled evidence switches change the required prefix record at matched depth and token difficulty. Targeted KV perturbations remove positions selected by future queries and leave target verification untouched. Direct cache reuse improves accepted length across every evaluated tree budget, with gains increasing as branches require more diverse prefix reads. At 32K context, duplicate prefix state nearly disappears, permitting a larger serving batch and 9.7% higher maximum-load throughput.

This work makes three contributions. First, it identifies prefix reconstruction as a distinct source of speculative readout error and characterizes the ambiguity left by finite query coverage. Second, it introduces OPSPEC, a target KV interface for causal-parallel drafting, together with readout

supervision and an exact split-attention implementation. Third, same-query comparisons, controlled evidence changes, cache interventions, and end-to-end serving experiments connect the isolated reconstruction error to later-step rejection and deployment efficiency.

2 Prefix Reconstruction Error

2.1 Query-Dependent Prefix Readouts

Let $P = x_{1:t}$ denote the verified prefix stored in the target cache. At target layer ℓ, its keys and values define

$$\mathcal{O}_P^\ell(q) = \mathrm{Attn}(q, K_P^\ell, V_P^\ell) = \mathrm{softmax}\left(\frac{qK_P^{\ell\top}}{\sqrt{d_h}}\right) V_P^\ell. \tag{1}$$

For fixed P, $\mathcal{O}_P^\ell$ maps a future query to the corresponding prefix readout. Candidate branches share K_P^ℓ and V_P^ℓ and differ through q. Hidden-conditioned drafting constructs a second map

$$\widetilde{\mathcal{O}}_P^\ell(q) = \mathrm{Attn}(q, \widetilde{K}_P^\ell, \widetilde{V}_P^\ell), \qquad (\widetilde{K}_P^\ell, \widetilde{V}_P^\ell) = g_\ell(H_P), \tag{2}$$

where H_P contains selected target hidden states and g_ℓ denotes feature fusion followed by draft-side KV projections.

For future node v, let $q_v^{T,\ell}$ be the target query and $\hat{q}_v^\ell$ the draft query. Target prefix readout is $r_v^{T,\ell} = \mathcal{O}_P^\ell(q_v^{T,\ell})$, and hidden-conditioned drafting returns $\hat{r}_v^\ell = \widetilde{\mathcal{O}}_P^\ell(\hat{q}_v^\ell)$. Suppose $q_v^{T,\ell}, \hat{q}_v^\ell \in \mathcal{Q}$ and $\mathcal{O}_P^\ell$ is L_q-Lipschitz on $\mathcal{Q}$. Adding and subtracting $\mathcal{O}_P^\ell(\hat{q}_v^\ell)$ yields

$$\|r_v^{T,\ell} - \hat{r}_v^\ell\| \le L_q \|q_v^{T,\ell} - \hat{q}_v^\ell\| + \epsilon_{\mathrm{rec}}^\ell(P, \mathcal{Q}), \tag{3}$$

$$\epsilon_{\mathrm{rec}}^\ell(P, \mathcal{Q}) = \sup_{q \in \mathcal{Q}} \|\mathcal{O}_P^\ell(q) - \widetilde{\mathcal{O}}_P^\ell(q)\|. \tag{4}$$

The first term in Equation 3 measures query error. The second records the change introduced by prefix reconstruction. Native target KV removes the second term for the selected layer up to numerical precision.

Same-query reconstruction error at node v is

$$\mathcal{E}_{\mathrm{rec}}(v, \ell) = \frac{\|\mathcal{O}_P^\ell(q_v^{T,\ell}) - \widetilde{\mathcal{O}}_P^\ell(q_v^{T,\ell})\|_2}{\|\mathcal{O}_P^\ell(q_v^{T,\ell})\|_2 + \varepsilon}. \tag{5}$$

Both maps receive the same target query. Any nonzero error originates in the prefix representation.

2.2 Finite-Query Ambiguity

Training encounters reconstructed KV through a finite collection of queries. Agreement on their outputs does not determine the response to every future query. Let $Q = \{q_1, \ldots, q_m\}$ be the observed query set. For fixed keys K, matrix $A_Q(K) \in \mathbb{R}^{m \times t}$ contains the corresponding attention rows, so observed readouts are $A_Q(K)V$. Any perturbation ΔV in the null space of $A_Q(K)$ satisfies

$$A_Q(K)(V + \Delta V) = A_Q(K)V. \tag{6}$$

Training queries cannot distinguish V from $V + \Delta V$. A new query $q_\star$ exposes the difference whenever $a_\star(K)\Delta V \neq 0$. For $t \ge 2$ and $q_\star \notin \mathrm{span}(Q)$, distinct key sets can preserve attention weights on Q and alter weights at $q_\star$.

Accurate token prediction on common trajectories does not imply recovery of the target prefix map. Supervision constrains the readouts it visits. Candidate histories that redirect attention to another part of the prefix probe regions left weakly determined by those trajectories. More varied training queries reduce this ambiguity. Native KV removes it from the prefix representation.

2.3 Branch-Dependent Prefix Readouts

Let $a_v^{T,\ell}$ denote target attention at node v, restricted and renormalized over prefix positions. For nodes $\mathcal{V}_k$ at depth k, variation in prefix use is

$$\mathcal{D}_{\mathrm{read}}(P, k) = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \mathbb{E}_{u \neq v \in \mathcal{V}_k} \mathrm{JSD}(a_u^{T,\ell}, a_v^{T,\ell}). \tag{7}$$

Low values indicate that candidate paths consult similar prefix locations. High values indicate that their histories have redirected attention toward different evidence.

When path-conditioned queries vary across branches, a depth-only state cannot represent them all. For random branch U and predictor $f(P,k)$ without branch history, squared error obeys

$$\inf_f \mathbb{E}[\|q_{U,k}^T - f(P,k)\|_2^2 \mid P,k] = \operatorname{tr}\operatorname{Var}(q_{U,k}^T \mid P,k). \tag{8}$$

The conditional mean attains the minimum, leaving cross-branch variance. Prefix reconstruction has little visible effect when candidate histories induce similar reads. Its effect grows once branch history redirects attention, particularly when $\mathcal{D}_{\text{read}}$ is high. Depth correlates with this event because longer paths permit more changes in relevant evidence. Depth itself does not determine which evidence a branch will retrieve.

3 OPSPEC

The analysis suggests one direct architectural intervention. OPSPEC keeps the pretrained causal-parallel branch model and replaces its hidden-derived full-prefix KV with the target cache. Query adapters connect existing branch states to target attention spaces. Readout supervision trains this interface, and direct page access avoids a second prefix cache.

3.1 DRAFT ARCHITECTURE

Every learned variant starts from the same five-layer JetSpec checkpoint for Qwen3-8B. Target parameters, shared embedding and output head, causal-parallel blocks, MLPs, and branch-local attention remain frozen. Conversion removes full-prefix hidden conditioning. A root projection summarizes the latest verified target states, and low-rank adapters map draft states into target attention spaces. Per-layer normalization and residual scaling support stable adaptation. No sequence-length-dependent hidden representation of the prefix remains in the drafter.

Draft layers correspond to target layers $\mathcal{L} = \{1, 9, 17, 25, 33\}$, matching the target views used by the baseline. One-to-one assignment places each query and its cached keys in one target attention space. Permuted assignments and shared-layer access are evaluated as controls.

3.2 BRANCH QUERIES AND NATIVE KV READOUTS

The newest committed position defines a root anchor, which initializes node v:

$$a_t = \operatorname{RMSNorm}\Big(W_a[h_t^{\ell_1}; \ldots; h_t^{\ell_5}]\Big), \tag{9}$$

$$z_v^0 = E(y_v^{\text{in}}) + a_t + e_{\text{depth}(v)}. \tag{10}$$

This constant-size anchor preserves short-range target information without retaining a prefix-length sequence.

At draft layer j, adapter $A_{Q,j}$ maps branch state into the hidden basis of target layer ℓ_j. Frozen target query projection, target Q-normalization, and rotary position embedding produce

$$\hat{q}_v^j = \operatorname{RoPE}_{p(v)}\Big(\operatorname{QNorm}^{T,\ell_j}\Big(W_Q^{T,\ell_j} A_{Q,j} \operatorname{Norm}(z_v^{j-1})\Big)\Big). \tag{11}$$

Cached target keys are consumed in their stored form. Target inference has already applied key projection, normalization, and rotary position encoding. Reapplying these transforms would change the attention geometry.

Frozen target projections and trainable rank-128 adapters generate local KV for speculative node u. Local memory contains u and its speculative ancestors. Prefix memory contains native KV for every cached token. Both segments are expressed in target attention coordinates.

3.3 EXACT SPLIT ATTENTION

Tree nodes share the prefix and retain short, path-specific ancestor histories. One kernel reads shared cache pages. A masked kernel reads branch-local KV. Let (O_P, λ_P) and (O_{L_v}, λ_{L_v}) denote their

outputs and log normalizers. Attention over their concatenation is recovered exactly by

$$O = e^{\lambda_P - \lambda} O_P + e^{\lambda_{L_v} - \lambda} O_{L_v}, \qquad \lambda = \mathrm{LSE}(\lambda_P, \lambda_{L_v}). \tag{12}$$

Both segments share one softmax denominator, which yields this identity and permits reuse of shared-prefix I/O across tree nodes (Yang et al., 2026; Yao et al., 2025).

The merged output returns to draft space through the frozen target output projection and a zero-initialized adapter:

$$u_v^j = z_v^{j-1} + \alpha_j B_j W_O^{T,\ell_j} O, \qquad z_v^j = u_v^j + \mathrm{MLP}_j(\mathrm{Norm}(u_v^j)). \tag{13}$$

Residual multiplier remains at 0.1 for 75 updates and grows linearly to one by update 250.

3.4 Operator Readout Distillation

Native KV fixes the prefix map, but branch states must still produce target-compatible queries. Forward KL supplies the output objective

$$\mathcal{L}_{\mathrm{FKL}} = T^2 \frac{\sum_v w_v D_{\mathrm{KL}}(p_v^T \| p_v^D)}{\sum_v w_v}. \tag{14}$$

For direct readout supervision, frozen target query q_v^{T,ℓ_j} is applied only to the verified prefix:

$$r_{P,v}^{T,j} = \mathrm{Attn}(q_v^{T,\ell_j}, K_P^{T,\ell_j}, V_P^{T,\ell_j}). \tag{15}$$

Draft readout $\hat{r}_{P,v}^j$ uses $\hat{q}_v^j$ and the same native KV. ORD minimizes

$$\mathcal{L}_{\mathrm{read}} = \frac{1}{5} \sum_j \frac{\sum_v w_v \| \mathrm{Norm}(\hat{r}_{P,v}^j) - \mathrm{sg}[\mathrm{Norm}(r_{P,v}^{T,j})] \|_2^2}{\sum_v w_v}, \tag{16}$$

$$\mathcal{L}_Z = \frac{1}{5} \sum_j \frac{\sum_v w_v (\log \hat{Z}_{P,v}^j - \mathrm{sg}[\log Z_{P,v}^{T,j}])^2}{\sum_v w_v}, \tag{17}$$

$$\mathcal{L}_q = \frac{1}{5} \sum_j \frac{\sum_v w_v (1 - \cos(\hat{q}_v^j, q_v^{T,\ell_j}))}{\sum_v w_v}. \tag{18}$$

Final loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{FKL}} + \mathcal{L}_{\mathrm{read}} + 0.05\mathcal{L}_Z + 0.1\mathcal{L}_q. \tag{19}$$

Readout matching receives the largest weight because different query vectors can induce similar prefix computations. Teacher-forced blocks cover the first 820 updates. During the final 180 updates, the current model generates candidate branches and target readouts supervise the visited states.

3.5 Memory and Execution

The hidden-conditioned baseline stores a fused 4096-dimensional BF16 prefix sequence and five layers of contextual draft KV with eight heads of dimension 128. This state occupies 28 KiB per prefix token. A reference implementation retaining all five target hidden taps occupies 60 KiB per token. OpSpec aliases target KV pages and retains only the root anchor and active tree states. Drafter-side prefix memory no longer grows with context length.

4 Experimental Setup

4.1 Models, Data, and Decoding

Qwen3-8B in non-thinking mode serves as the target. Every learned model starts from the same JetSpec checkpoint. Target parameters and the causal-parallel backbone remain frozen. Training updates prefix-interface adapters, root anchor, and normalizations. Residual multipliers follow a fixed schedule. We sample 128,000 examples from the JetSpec training mixture, preserve the proportions

of Nemotron Post-Training Dataset V2 and CodeAlpaca, and regenerate responses with Qwen3-8B. Global batch 128 gives 1,000 optimizer updates. AdamW uses learning rate 3×10^{-4}, weight decay 0.1, BF16, 3% warm-up, and cosine decay. Learned model rows report three-seed aggregates.

Candidate trees have maximum depth 16, width 10, and budgets $B \in \{16, 32, 64, 128, 256\}$. Tree construction and target verification are identical across methods. Evaluation covers GSM8K, MATH-500, AIME 2025, HumanEval, MBPP, LiveCodeBench, and MT-Bench. Serving tests use 2K and 32K prompts on H100 GPUs at concurrency one and sixteen.

The primary metric is average accepted length τ, defined as committed output tokens per target verification round. System measurements include speedup over target autoregressive decoding, committed tokens per verified node, peak memory, proposal latency, and maximum feasible batch.

4.2 Baselines and Controls

JetSpec-N is the main baseline. It matches OPSPEC in normalization, root anchor, parameter budget, data, and update schedule, but retains hidden-derived prefix KV. DDTree and original JetSpec checkpoint run in the same decoding engine. Additional controls use equal-parameter adapters, target-dimensional surrogate KV, permuted target-layer mapping, shared-layer KV, no root anchor, and independent normalization of prefix and branch attention.

Prefix-source analysis uses one switchable checkpoint per seed for all four conditions, which removes between-condition checkpoint variation. Each training batch selects native or reconstructed KV with equal probability. At evaluation, nodes retain their branch-specific states or receive the mean state at the same depth. Difference-in-differences measures how direct cache access changes once the query reflects branch history.

4.3 Evidence Switching and Cache Interventions

Evidence Switching embeds 32 labeled records among natural-language distractors. Continuations emit selected values in a prescribed order. Same-read examples use one record throughout. Early-switch examples change the required record within two future positions. Late-switch examples change it after at least six positions. Branch-switch examples assign different records to competing high-probability branches. Conditions are matched by target entropy, output length, record position, and token frequency.

For future node v, future-specific evidence is

$$E_v = \mathrm{TopM}(a_v^T) \setminus \mathrm{TopM}(a_{\mathrm{root}}^T). \tag{20}$$

Draft-time interventions mask E_v, shuffle its values, or mask random and recency-matched controls. Target verification uses the unmodified cache, preserving target output.

5 Results

5.1 Same-Query Reconstruction Error

Applying one target query to both prefix representations removes uncertainty in the future state. Hidden-derived KV yields 0.186 readout NRMSE and 0.943 cosine similarity to target readout. Native KV yields 0.0002 NRMSE and 0.9999 cosine similarity. A target-dimensional surrogate lowers NRMSE to 0.143, yet its error remains orders of magnitude above direct cache reuse.

Discrepancy grows when candidate branches use the prefix differently. Median native-surrogate readout difference rises from 0.112 in the lowest readout-diversity quartile to 0.247 in the highest. A mixed-effects rejection model controls for depth, target entropy, token likelihood, and branch rank. The standardized coefficient of same-query reconstruction error is 1.74, with 95% confidence interval $[1.31, 2.18]$. Readout diversity has coefficient 0.91. Adding both quantities improves held-out log likelihood by 4.6% and reduces the remaining depth coefficient by 38.9%. Figure 2 links this isolated error to the controlled accepted-length and tree-budget effects.

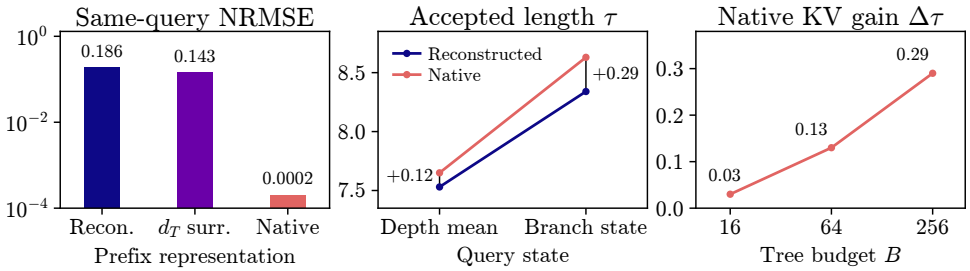


Figure 2: Native KV improves readouts and acceptance. Left: same-query NRMSE. Center: accepted length under depth-mean and branch-specific states. Right: gain versus tree budget B.

5.2 Evidence Reactivation and Rejection

Evidence Switching changes the prefix record required by a continuation and keeps the verified prefix fixed. Same-read examples lose 4.3 percentage points of acceptance from positions one to eight. Changing the record within the first two positions produces an immediate 13.8-point drop. Native KV offers little improvement when every position reads the same record and markedly more after the evidence demand changes. Branch-switch examples show the largest gain because plausible paths require different records at the same depth.

Table 1: Evidence-switching and cache-intervention results at $B = 256$.

Condition	Evidence change	Gain	Intervention	Drop
Same-read	None	0.9%	Future-specific mask	8.6 pp
Early-switch	Positions 1–2	4.7%	Future-specific shuffle	7.2 pp
Late-switch	After position 6	4.1%	Random-position mask	1.4 pp
Branch-switch	Differs by path	6.4%	Recency-matched mask	1.9 pp

After matching target entropy, output length, record position, and token frequency, standardized coefficient of prefix-attention shift is 1.61. Residual coefficient of draft depth is 0.23. Evidence demand accounts for the abrupt decline even when the switch occurs near the root.

Cache interventions provide a direct causal test. Future-specific positions receive high attention from target future query and low attention from current query. Masking them lowers node acceptance by 8.6 percentage points, and shuffling their values lowers it by 7.2 points. Count- and recency-matched controls reduce acceptance by 1.4 and 1.9 points. Targeted masking removes 71.4% of OpSpec's gain in the highest readout-diversity quartile and 16.8% in the lowest. The full cache remains available to target verification, so interventions change proposal quality without changing final output.

5.3 Decoding Performance

Table 2 reports Qwen3-8B results. Small trees produce similar results because accepted paths remain close to the current target state. At budget 256, OpSpec raises average accepted length from 8.34 to 8.63 and speedup from 7.33× to 7.54×. Accepted-length gains reach 5.1% on MATH-500, 4.7% on AIME25, 4.0% on LiveCodeBench, 2.4% on MBPP, and 1.8% on MT-Bench. Temperature-one decoding retains gains of 2.8% in accepted length and 2.2% in throughput.

The controlled query-state intervention shows when prefix source affects decoding. Native rather than reconstructed KV raises τ from 7.53 to 7.65 with depth-wise mean states and from 8.34 to 8.63 with branch-specific states. Difference-in-differences is 0.17 accepted tokens, with interval $[0.06, 0.28]$ and $p = 0.0024$. Excess gain increases from 0.05 in the lowest readout-diversity quartile to 0.31 in the highest. Equal-parameter adapters improve τ by 0.6%, and target-dimensional surrogate KV remains 2.2% below direct cache reuse. Appendix A reports all four cells.

Table 2: Qwen3-8B results. τ is the seven-task geometric mean. Prefix payload excludes target KV, root anchor, and active tree.

Method	τ_{16}	τ_{64}	τ_{256}	Speedup$_{256}$	Payload (KiB/token)
DDTree	5.80	5.93	7.53	6.64×	28
JetSpec	5.80	6.27	8.32	7.31×	28–60
JetSpec-N	5.84	6.32	8.34	7.33×	28
OPSPEC	**5.87**	**6.45**	**8.63**	**7.54×**	**0**

5.4 TREE BUDGET AND READOUT DIVERSITY

Tree-budget scaling provides an independent check on the proposed mechanism. Native KV adds 0.03 accepted tokens at budget 16, 0.13 at budget 64, and 0.29 at budget 256. Relative improvement grows from 0.5% to 3.5% across the same range. Node budget is identical in each comparison, and committed tokens per verified node increase from 0.0326 to 0.0337 at budget 256. Direct cache access uses a fixed candidate budget more effectively. Verification work is unchanged.

Task-level results follow the same pattern. MATH-500 and AIME 2025 show the largest improvements, followed by LiveCodeBench. Both reasoning benchmarks frequently sustain a candidate path for several steps before reusing an earlier constraint or variable. LiveCodeBench exhibits a similar pattern for identifiers. MBPP and MT-Bench show smaller gains, where shorter accepted paths leave fewer later nodes at which prefix fidelity can affect the outcome. Persistence under temperature-one decoding indicates that direct cache access improves candidate distributions beyond greedy top-one agreement.

5.5 READOUT SUPERVISION ABLATIONS

Once native KV fixes prefix computation, remaining gap depends on query produced by candidate path. Oracle target queries reach $\tau = 9.17$. Native KV with depth-wise mean branch state gives lower endpoint 7.65. Forward KL reaches 8.41 and recovers 50.0% of this 1.52-token interval. Query-direction supervision reaches 8.46. Prefix-readout matching reaches 8.56, log-normalizer matching reaches 8.59, and on-policy ORD reaches 8.63. Query cosine rises from 0.772 to 0.879 as readout NRMSE falls from 0.219 to 0.132.

Table 3: Training ablations at $B = 256$ under matched 1,000-update schedules.

Objective	Query cosine	Readout NRMSE	τ	Speedup
Forward KL	0.772	0.219	8.41	7.36×
+ query direction	0.844	0.198	8.46	7.39×
+ prefix readout	0.861	0.153	8.56	7.48×
+ log normalizer	0.867	0.144	8.59	7.50×
+ on-policy branches	0.879	0.132	8.63	7.54×
Oracle target query	1.000	0.000	9.17	–

Removing the root anchor lowers first-step acceptance by 2.7 percentage points and total τ by 1.5%. Permuted target-layer assignment costs 2.0%. Five aligned target layers outperform one shared layer by 1.3%. Independent normalization of prefix and branch attention loses 0.9% relative to exact merging. Every control weakens performance, excluding parameter count and generic normalization as sufficient explanations.

5.6 MEMORY AND SERVING EFFICIENCY

An unfused native-prefix prototype adds 5.6% to proposal latency. Removing hidden fusion and contextual KV projection recovers 4.2%. Fused split attention yields net reductions of 2.4% at 2K context and 5.8% at 32K. Figure 3 reports memory, batch capacity, and throughput.

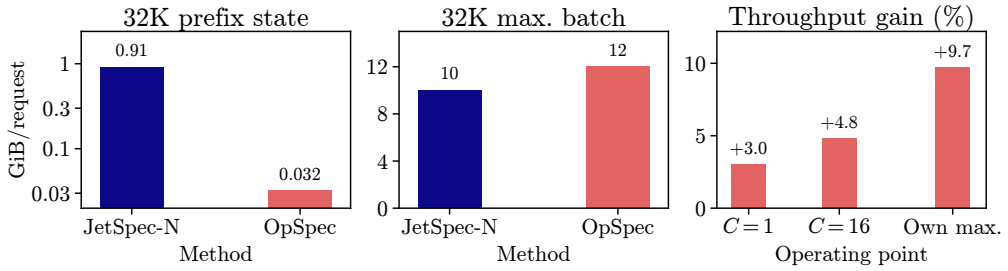


Figure 3: H100 serving at 32K context. OPSPEC uses 96.5% less drafter state, supports a 20% larger batch, and improves throughput by up to 9.7% over JetSpec-N.

Budget 128 maximizes throughput at moderate concurrency, whereas budget 256 maximizes offline accepted length.

6 RELATED WORK

Speculative decoding combines inexpensive proposals with exact target correction (Leviathan et al., 2023; Chen et al., 2023; Kim et al., 2023). Distillation, online adaptation, optimal-transport acceptance, retrieval, recurrent drafting, and phrase reuse improve the proposal distribution (Zhou et al., 2024; Liu et al., 2024b; Sun et al., 2023; He et al., 2024; Cheng et al., 2024; Zhao et al., 2024). These methods change candidate generation without altering target verification.

Head-based and self-speculative methods reuse target computation. Medusa, Hydra, EAGLE, EAGLE-2, and EAGLE-3 employ auxiliary heads, sequential head dependencies, feature prediction, and adaptive trees (Cai et al., 2024; Ankner et al., 2024; Li et al., 2024b;c; 2025). LayerSkip and Kangaroo draft from early exits (Elhoushi et al., 2024; Liu et al., 2024a). Multi-token prediction, blockwise validation, Jacobi iteration, and Lookahead Decoding compute several future positions in parallel (Gloeckle et al., 2024; Stern et al., 2018; Santilli et al., 2023; Fu et al., 2024). Tree systems improve coverage with shared verification and adaptive topology (Miao et al., 2024; Chen et al., 2024; Wang et al., 2025; Zhong et al., 2024; Jeon et al., 2024; Ringel & Romano, 2026; Hu et al., 2026).

Target KV access has appeared in several drafters. GliDe uses cross-attention over target cache, LongSpec combines target KV with local attention, and KVShot compares hidden, KV, and hybrid conditioning within one autoregressive pipeline (Du et al., 2024; Yang et al., 2026; Liu et al., 2026). DFlare assigns distinct target-layer views to deeper draft layers (Zhang et al., 2026). Our study examines the prefix reconstruction embedded in a causal-parallel head. Same-query evaluation isolates its readout error, and OPSPEC removes that reconstruction from the drafting path.

KV cache research primarily reduces storage and access cost through eviction, streaming windows, query-aware selection, and quantization (Zhang et al., 2023; Liu et al., 2023; Xiao et al., 2024; Li et al., 2024a; Cai et al., 2025; Tang et al., 2024; Liu et al., 2024c; Hooper et al., 2024). PagedAttention, FlashAttention, FlashAttention-2, FlashInfer, and DeFT reduce fragmentation or memory traffic (Kwon et al., 2023; Dao et al., 2022; Dao, 2024; Ye et al., 2025; Yao et al., 2025). OPSPEC changes which prefix representation the drafter reads. Existing cache kernels then implement that access efficiently.

7 CONCLUSION

Target inference already stores KV for every verified token, yet hidden-conditioned drafters commonly rebuild another prefix cache. Same-query evaluation shows that this reconstruction changes future prefix readouts and becomes consequential when a candidate path returns to earlier evidence. OPSPEC removes the reconstructed path, maps branch states into target attention spaces, and reads native KV directly. Readout supervision improves the remaining query, and exact split attention reuses cached pages without duplication. Direct target-cache access improves accepted length and throughput and removes nearly all drafter-side prefix state at long context.

AI USE STATEMENT

We used large language models (LLMs) to improve grammar and correct spelling errors in the manuscript. The authors take full responsibility for the final content of this work.

REFERENCES

Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. Hydra: Sequentially-dependent draft heads for Medusa decoding. In *First Conference on Language Modeling*, 2024.

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, pp. 5209–5235, 2024.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025.

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.

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding. In *Forty-Third International Conference on Machine Learning*, 2026.

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. Sequoia: Scalable and robust speculative decoding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 129531–129563, 2024.

Yunfei Cheng, Aonan Zhang, Xuanyu Zhang, Chong Wang, and Yi Wang. Recurrent drafter for fast speculative decoding in large language models. *arXiv preprint arXiv:2403.09919*, 2024.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022.

Cunxiao Du, Jing Jiang, Xu Yuanchen, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, and Yang You. GliDe with a CaPE: A low-hassle method to accelerate speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 11704–11720, 2024.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A. Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642, 2024.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of LLM inference using lookahead decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 14060–14079, 2024.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 15706–15734, 2024.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason Lee, and Di He. REST: Retrieval-based speculative decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1582–1595, 2024.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 1270–1303, 2024.

Lanxiang Hu, Zhaoxiang Feng, Yulun Wu, Haoran Yuan, Yujie Zhao, Yu-Yang Qian, Bojun Wang, Peng Zhao, Daxin Jiang, Yibo Zhu, Tajana Rosing, and Hao Zhang. JetSpec: Breaking the scaling ceiling of speculative decoding with parallel tree drafting. *arXiv preprint arXiv:2606.18394*, 2026.

Wonseok Jeon, Mukul Gagrani, Raghavv Goel, Junyoung Park, Mingu Lee, and Christopher Lott. Recursive speculative decoding: Accelerating LLM inference via sampling without replacement. In *ICLR 2024 Workshop on Large Language Models for Agents*, 2024.

Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W. Mahoney, Amir Gholami, and Kurt Keutzer. Speculative decoding with Big Little Decoder. In *Advances in Neural Information Processing Systems*, volume 36, pp. 39236–39256, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

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, pp. 19274–19286, 2023.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970, 2024a.

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, pp. 28935–28948, 2024b.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7421–7432, 2024c.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In *Advances in Neural Information Processing Systems*, volume 38, pp. 136737–136756, 2025.

Fangcheng Liu, Yehui Tang, Zhenhua Liu, Yunsheng Ni, Duyu Tang, Kai Han, and Yunhe Wang. Kangaroo: Lossless self-speculative decoding for accelerating LLMs via double early exiting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 11946–11965, 2024a.

Tianyu Liu, Yuhao Shen, Xinyi Hu, Baolin Zhang, Hengxin Zhang, Jun Dai, Jun Zhang, Shuang Ge, Lei Chen, Yue Li, and MingCheng Wan. When hidden states drift: Can KV caches rescue long-range speculative decoding? *arXiv preprint arXiv:2604.26412*, 2026.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 31131–31146, 2024b.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52342–52364, 2023.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32332–32344, 2024c.

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*, pp. 932–949, 2024.

Liran Ringel and Yaniv Romano. Accelerating speculative decoding with block diffusion draft trees. *arXiv preprint arXiv:2604.12989*, 2026.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodolà. Accelerating transformer inference for translation via parallel decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12336–12355, 2023.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*, volume 31, pp. 10086–10095, 2018.

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, pp. 30222–30242, 2023.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 47901–47911, 2024.

Jikai Wang, Yi Su, Juntao Li, Qingrong Xia, Zi Ye, Xinyu Duan, Zhefeng Wang, and Min Zhang. OPT-Tree: Speculative decoding with adaptive draft tree structure. *Transactions of the Association for Computational Linguistics*, 13:188–199, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Penghui Yang, Cunxiao Du, Fengzhuo Zhang, Haonan Wang, Tianyu Pang, Chao Du, and Bo An. LongSpec: Long-context lossless speculative decoding with efficient drafting and verification. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1826–1844, 2026.

Jinwei Yao, Kaiqi Chen, Kexun Zhang, Jiaxuan You, Binhang Yuan, Zeke Wang, and Tao Lin. DeFT: Decoding with flash tree-attention for efficient tree-structured LLM inference. In *International Conference on Learning Representations*, volume 2025, pp. 66657–66688, 2025.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. FlashInfer: Efficient and customizable attention engine for LLM inference serving. In *Proceedings of Machine Learning and Systems*, volume 7, 2025.

Jiebin Zhang, Zhenghan Yu, Song Liu, Eugene J. Yu, Zheng Li, Dawei Zhu, Jiangshan Duo, Weimin Xiong, Yifan Song, Guanghua Yu, Jianchen Zhu, and Sujian Li. DFlare: Scaling up draft capacity for block diffusion speculative decoding. *arXiv preprint arXiv:2606.02091*, 2026.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710, 2023.

Weilin Zhao, Yuxiang Huang, Xu Han, Wang Xu, Chaojun Xiao, Xinrong Zhang, Yewei Fang, Kaihuo Zhang, Zhiyuan Liu, and Maosong Sun. Ouroboros: Generating longer drafts phrase by phrase for faster speculative decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13378–13393, 2024.

Shuzhang Zhong, Zebin Yang, Ruihao Gong, Runsheng Wang, Ru Huang, and Meng Li. ProPD: Dynamic token tree pruning and generation for LLM parallel decoding. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 202:1–202:8, 2024.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, volume 2024, pp. 32011–32050, 2024.

A Prefix-Source Intervention

A.1 Factorial Design

Each seed uses one switchable checkpoint for all four cells. Training samples native and reconstructed prefix KV with equal probability. Evaluation crosses prefix source with branch-specific or depth-mean query states. Checkpoint parameters stay fixed. Reported cells aggregate three seeds.

Prefix-map error is recomputed with the query induced by each query-state condition. Native replay uses the target cache as its reference. Values near zero reflect numerical replay at the displayed precision.

Table 4: Prefix-source intervention at $B = 256$.

Prefix KV	Query state	Induced-query NRMSE	τ	Speedup	Efficiency
Reconstructed	Depth mean	0.186	7.53	$6.64\times$	0.0294
Native	Depth mean	< 0.001	7.65	$6.73\times$	0.0299
Reconstructed	Branch-specific	0.194	8.34	$7.33\times$	0.0326
Native	Branch-specific	< 0.001	8.63	$7.54\times$	0.0337

A.2 Cell Effects

Native KV raises accepted length by $7.65 - 7.53 = 0.12$ under depth-mean states. Branch-specific states raise the same source effect to $8.63 - 8.34 = 0.29$. Difference-in-differences equals

$$\Delta_{\mathrm{DiD}} = (8.63 - 8.34) - (7.65 - 7.53) = 0.17. \tag{21}$$

Interval $[0.06, 0.28]$ and $p = 0.0024$ refer to the interaction estimate.

Speedup changes by $0.09\times$ under depth averaging and by $0.21\times$ under branch-specific states. Efficiency changes by 0.0005 and 0.0011 committed tokens per verified node. Larger changes under branch-specific states agree with the readout-diversity analysis around Equation 7.

B Reproducibility Details

B.1 Model and Trainable Components

Qwen3-8B runs in non-thinking mode as the target model. Every learned condition begins from the same five-layer JetSpec checkpoint. Target parameters, shared embedding, output head, causal-parallel blocks, MLPs, and branch-local attention remain frozen. Optimization updates the prefix-interface adapters, root-anchor path, and per-layer normalizations. Residual multipliers follow the schedule below.

Five draft layers map to target layers

$$\mathcal{L} = \{1, 9, 17, 25, 33\}.$$

One-to-one assignment places each draft query in one target attention space. Permuted-layer and shared-layer conditions alter only the assignment. Rank-128 adapters supply the trainable interface used by query and local KV paths.

JetSpec-N uses the same normalization, root anchor, parameter budget, data, and update schedule as OPSPEC. Hidden-derived prefix KV remains active in JetSpec-N. DDTree and the original JetSpec checkpoint run in the same decoding engine.

B.2 DATA AND OPTIMIZATION

Training uses 128,000 examples sampled from the JetSpec mixture. Sampling preserves the mixture proportions of Nemotron Post-Training Dataset V2 and CodeAlpaca. Qwen3-8B regenerates responses used for target-aligned supervision. Global batch 128 yields 1,000 optimizer updates.

AdamW uses learning rate 3×10^{-4}, weight decay 0.1, BF16 arithmetic, 3% warm-up, and cosine decay. Learned model rows report three-seed aggregates. Every learned row in Table 3 follows the same 1,000-update budget.

Residual multiplier stays at 0.1 through update 75 and grows linearly to one by update 250. Full ORD training uses teacher-forced candidate blocks through update 820. Current-model branches supply states during its final 180 updates.

Table 5: Training stages for the full ORD configuration.

Updates	Candidate states	Residual multiplier
1–75	Teacher-forced	0.1
76–250	Teacher-forced	Linear $0.1 \to 1$
251–820	Teacher-forced	1
821–1000	Current drafter	1

B.3 SUPERVISION

Frozen target computation supplies the token distribution, target query, prefix-only readout, and prefix log normalizer for every supervised node. Draft computation produces matching quantities from branch state and native prefix KV. Loss terms follow

$$\mathcal{L} = \mathcal{L}_{\mathrm{FKL}} + \mathcal{L}_{\mathrm{read}} + 0.05\mathcal{L}_Z + 0.1\mathcal{L}_q.$$

Forward KL supervises token probabilities. Readout and log-normalizer losses respectively match normalized prefix outputs and prefix attention mass before normalization. Query-direction loss aligns draft and target directions. Stop-gradient is applied to target-side signals.

Every node-weighted term uses denominator $\sum_v w_v$. Common rescaling of node weights leaves each normalized term unchanged. Full ORD uses current-drafter states during its final 180 updates. Target parameters and cached prefix tensors stay fixed.

B.4 TREE DECODING AND METRICS

Candidate trees use maximum depth 16 and width 10. Evaluated node budgets are

$$B \in \{16, 32, 64, 128, 256\}.$$

Tree construction, candidate budget, target verification, and acceptance rule remain matched within every method comparison.

Evaluation covers GSM8K, MATH-500, AIME 2025, HumanEval, MBPP, LiveCodeBench, and MT-Bench. For task d, let $N_{\mathrm{commit},d}$ be committed output tokens and let $R_{\mathrm{verify},d}$ be target verification rounds. Accepted length is

$$\tau_d = \frac{N_{\mathrm{commit},d}}{R_{\mathrm{verify},d}}.$$

Table 2 reports the geometric mean of the seven task-level values. For N_{node} verified candidate nodes, efficiency is

$$\eta = \frac{N_{\mathrm{commit}}}{N_{\mathrm{node}}}.$$

Speedup uses target autoregressive decoding as reference. System measurements report incremental drafter memory, proposal latency, throughput, and maximum feasible batch. All compared systems require target KV, so incremental drafter memory omits it.

C Implementation Details

C.1 Cache Ownership

Target inference owns the verified-prefix cache. At each assigned target layer ℓ_j, OpSpec reads all cache pages covering the verified prefix through read-only references. No token-level eviction or sparse prefix selection is used.

Speculative local KV contains each node and its speculative ancestors. Active-tree storage grows with node budget B. Beyond target KV, drafter storage contains the root anchor and active-tree state.

C.2 Query Coordinates

Newest committed target states form the root anchor

$$a_t = \mathrm{RMSNorm}\Big(W_a \,\mathrm{concat}(h_t^{\ell_1}, \ldots, h_t^{\ell_5})\Big).$$

Node input combines token embedding, root anchor, and depth embedding. Adapter $A_{Q,j}$ maps the branch state into the hidden basis of target layer ℓ_j. Frozen target query projection and target Q-normalization then produce a query. Rotary position embedding uses candidate position $p(v)$.

Cached keys already contain target key projection, target normalization, and rotary position encoding. Reapplying any of these transforms changes score geometry. Prefix-page reads consume stored keys and values without modification.

Frozen target projections and trainable rank-128 adapters generate speculative local KV. Position indices follow each candidate path. Prefix and local segments share target attention coordinates at layer ℓ_j.

C.3 Draft-Layer Execution

One draft layer processes node v in these steps:

1. Form branch query $\hat{q}_v^j$ from z_v^{j-1}.
2. Read target-cache pages and return prefix output O_P with log normalizer λ_P.
3. Generate local KV for v and its speculative ancestors.
4. Apply the path mask and return local output O_{L_v} with log normalizer λ_{L_v}.
5. Merge both outputs with one softmax denominator.
6. Apply frozen target output projection and the trainable output adapter.
7. Propagate the branch state through the frozen causal-parallel block and MLP.

Prefix pages can be reused across nodes because every branch shares the verified prefix. Local masks retain path dependence. Target verification receives the candidate tree through the same engine and acceptance rule used by baselines.

C.4 Numerical Scope of the Merge

Segmented attention equals attention over concatenated valid positions in real arithmetic. Floating-point reduction order can change the final bits. Reported native replay NRMSE of 0.0002 indicates agreement at the displayed precision. Comparison fixes query, head, mask, and positional coordinates. Both segment outputs retain target attention coordinates before the frozen output projection. Merge consumes only the two output-normalizer pairs. Appendix F states the stabilized merge and empty-segment convention.

D Diagnostic Design

D.1 Same-Query Readout

For candidate node v, target query $q_v^{\mathrm{T},\ell}$ is applied to native target KV and reconstructed KV. Both maps receive the same query. Readout differences come from the two prefix maps at that query. Calculation uses target queries directly, independent of candidate selection and draft-query prediction.

Normalized error follows Equation 5. Cosine similarity measures readout direction. Table 6 reports aggregate NRMSE and cosine similarity.

Table 6: Same-query prefix-readout measurements.

Prefix representation	NRMSE ↓	Cosine ↑
Reconstructed KV	0.186	0.943
Target-dimensional surrogate	0.143	–
Native target KV	0.0002	0.9999

Target dimensionality lowers reconstructed NRMSE from 0.186 to 0.143. Native replay reaches NRMSE 0.0002 and cosine similarity 0.9999.

D.2 Readout Diversity and Rejection

Target attention is restricted to prefix positions and renormalized before evaluating Equation 7. Pairwise Jensen–Shannon divergence is averaged across selected target layers and node pairs at one depth. Low and high readout-diversity quartiles have median native-surrogate readout differences of 0.112 and 0.247.

Reported mixed-effects analysis controls for depth, target entropy, target token likelihood, and branch rank. Standardized coefficient for reconstruction error is 1.74 with 95% interval $[1.31, 2.18]$. Readout diversity has coefficient 0.91. Adding both predictors improves held-out log likelihood by 4.6% and reduces the remaining depth coefficient by 38.9%. Reported coefficients describe conditional association. Cache interventions below apply targeted perturbations to future-specific prefix positions.

D.3 Evidence Switching

Each prefix contains 32 labeled records placed among natural-language distractors. Continuations emit selected values in a prescribed order. Conditions vary the time or branch at which required evidence changes. Matching uses target entropy, output length, record position, and token frequency.

Table 7: Native KV gains by evidence condition at $B = 256$.

Condition	Evidence change	Native KV gain
Same-read	None	0.9%
Early-switch	Positions 1–2	4.7%
Late-switch	After position 6	4.1%
Branch-switch	Differs by path	6.4%

Same-read examples lose 4.3 percentage points of acceptance from positions one through eight. Early switching produces an immediate 13.8-point drop.

D.4 Cache Interventions

Future-specific evidence for node v is

$$E_v = \mathrm{TopM}(a_v^{\mathrm{T}}) \setminus \mathrm{TopM}(a_{\mathrm{root}}^{\mathrm{T}}).$$

Here $\mathrm{TopM}(a)$ returns the M prefix positions with largest attention weight. Set E_v contains positions that receive high future-query attention and lie outside the root query's top set. Drafter-side variants

mask E_v, shuffle its values, or mask count- and recency-matched control positions. Target verification keeps its original cache.

Table 8: Acceptance drops under drafter-side cache interventions.

Intervention	Acceptance drop
Future-specific mask	8.6 pp
Future-specific shuffle	7.2 pp
Random-position mask	1.4 pp
Recency-matched mask	1.9 pp

Targeted masking removes 71.4% of OPSPEC's gain in the highest readout-diversity quartile and 16.8% in the lowest.

D.5 REPORTED BUDGET CHECKPOINTS

Evaluation also includes $B = 32$ and $B = 128$. Table 9 lists the three checkpoints shown in Figure 2.

Table 9: Accepted length at reported tree budgets.

B	JetSpec-N	OPSPEC	Absolute gain	Relative gain
16	5.84	5.87	0.03	0.5%
64	6.32	6.45	0.13	2.1%
256	8.34	8.63	0.29	3.5%

Task-level relative gains at $B = 256$ are 5.1% on MATH-500, 4.7% on AIME 2025, 4.0% on LiveCodeBench, 2.4% on MBPP, and 1.8% on MT-Bench. Temperature-one decoding retains gains of 2.8% in accepted length and 2.2% in throughput.

D.6 READOUT SUPERVISION ABLATIONS

Learned rows in Table 3 add objectives cumulatively. Forward KL reaches $\tau = 8.41$. Adding query-direction, prefix-readout, log-normalizer, and on-policy supervision raises τ to 8.46, 8.56, 8.59, and 8.63. Oracle target queries are used only at evaluation and reach 9.17.

Removing the root anchor lowers first-step acceptance by 2.7 percentage points and total accepted length by 1.5%. Permuted target-layer assignment costs 2.0%. Five aligned target layers outperform one shared layer by 1.3%. Independent prefix and local normalization loses 0.9% relative to the shared-denominator merge.

E READOUT DERIVATIONS

E.1 NOTATION

Derivations fix one verified prefix and one target layer. Queries use column form

$$a_K(q) = \mathrm{softmax}\left(\frac{Kq}{\sqrt{d_h}}\right), \qquad \mathcal{O}_{K,V}(q) = V^\top a_K(q). \tag{22}$$

Equation 22 is the transpose form of Equation 1. Symbols K and V abbreviate native prefix tensors. Key dimension is d_h, and d_v denotes per-head value dimension. Tildes mark reconstructed tensors. Random variable U denotes candidate branch identity. Symbol B denotes tree budget. For vectors, $\|\cdot\|_2$ denotes Euclidean norm. For matrices, it denotes spectral norm.

E.2 Continuity of Prefix Attention

Proposition E.1 (Lipschitz bound). *For fixed $K \in \mathbb{R}^{t\times d_h}$ and $V \in \mathbb{R}^{t\times d_v}$, all $q, q' \in \mathbb{R}^{d_h}$ satisfy*

$$\|\mathcal{O}_{K,V}(q) - \mathcal{O}_{K,V}(q')\|_2 \le \frac{\|K\|_2\|V\|_2}{2\sqrt{d_h}}\|q - q'\|_2. \tag{23}$$

Proof. Let σ denote softmax and set $p = \sigma(s)$. Its Jacobian is

$$J_\sigma(s) = \operatorname{diag}(p) - pp^\top.$$

Row i has diagonal entry $p_i(1-p_i)$. Sum of absolute off-diagonal entries in that row has the same value. Gershgorin's theorem gives

$$\|J_\sigma(s)\|_2 \le 2\max_i p_i(1-p_i) \le \frac{1}{2}.$$

Set $\gamma(\theta) = q' + \theta(q - q')$. Integrating along this path gives

$$\mathcal{O}_{K,V}(q) - \mathcal{O}_{K,V}(q') = \int_0^1 V^\top J_\sigma\left(\frac{K\gamma(\theta)}{\sqrt{d_h}}\right)\frac{K(q-q')}{\sqrt{d_h}}\,d\theta.$$

Submultiplicativity and the Jacobian bound give Equation 23. □

E.3 Readout Error Decomposition

Add and subtract $\mathcal{O}_P^\ell(\hat{q}_v^\ell)$ in the full readout error

$$\begin{aligned} r_v^{\mathrm{T},\ell} - \hat{r}_v^\ell &= \mathcal{O}_P^\ell(q_v^{\mathrm{T},\ell}) - \mathcal{O}_P^\ell(\hat{q}_v^\ell) \\ &\quad + \mathcal{O}_P^\ell(\hat{q}_v^\ell) - \widetilde{\mathcal{O}}_P^\ell(\hat{q}_v^\ell). \end{aligned} \tag{24}$$

Assume $q_v^{\mathrm{T},\ell}, \hat{q}_v^\ell \in \mathcal{Q}$ and let $\mathcal{O}_P^\ell$ be L_q-Lipschitz on $\mathcal{Q}$. Triangle inequality bounds the first difference by $L_q\|q_v^{\mathrm{T},\ell} - \hat{q}_v^\ell\|_2$. By definition, $\epsilon_{\mathrm{rec}}^\ell(P, \mathcal{Q})$ bounds the second difference. Equation 3 follows.

Same-query evaluation bypasses the learned draft query and evaluates both prefix operators at $q_v^{\mathrm{T},\ell}$. Query displacement vanishes. Numerator in Equation 5 equals prefix-map discrepancy at the selected query.

Proposition E.2 (Two-sided readout bound). *Let $\mathcal{O}$ be L_q-Lipschitz and define the unnormalized same-query discrepancy*

$$e_{\mathrm{abs}}(q) = \|\mathcal{O}(q) - \widetilde{\mathcal{O}}(q)\|_2.$$

For target query q^{T} and draft query $\hat{q}$,

$$\begin{aligned} \max\{0, e_{\mathrm{abs}}(\hat{q}) - L_q\|q^{\mathrm{T}} - \hat{q}\|_2\} &\le \|\mathcal{O}(q^{\mathrm{T}}) - \widetilde{\mathcal{O}}(\hat{q})\|_2 \\ &\le e_{\mathrm{abs}}(\hat{q}) + L_q\|q^{\mathrm{T}} - \hat{q}\|_2. \end{aligned} \tag{25}$$

Proof. Adding $\mathcal{O}(\hat{q})$ and applying triangle inequality gives the upper bound. Applying reverse triangle inequality gives the lower bound. Lipschitz continuity controls $\|\mathcal{O}(q^{\mathrm{T}}) - \mathcal{O}(\hat{q})\|_2$ in both directions. □

When $\hat{q} = q^{\mathrm{T}}$, $e_{\mathrm{abs}}(q^{\mathrm{T}})$ is the unnormalized numerator in Equation 5. Reported NRMSE divides it by $\|\mathcal{O}(q^{\mathrm{T}})\|_2 + \varepsilon$.

Remark E.3. Positive same-query error proves that two readout maps disagree at the measured query. Zero measured error establishes agreement at that query only. Raw KV tensors need not be unique.

E.4 Finite-Query Value Ambiguity

For observed queries $Q = \{q_1, \ldots, q_m\}$, matrix $A_Q(K)$ contains one attention row per query. Observed readouts form $A_Q(K)V$.

Proposition E.4 (Unobserved value subspace). *Let $r = \mathrm{rank}(A_Q(K))$. Set*

$$\mathcal{N}_Q = \left\{ \Delta V \in \mathbb{R}^{t \times d_v} \mid A_Q(K)\Delta V = 0 \right\}.$$

Its dimension is

$$\dim(\mathcal{N}_Q) = (t - r)d_v. \tag{26}$$

For fixed K, observed readouts determine V uniquely if and only if $r = t$.

Proof. Every column of ΔV belongs to $\mathrm{Null}(A_Q(K))$, whose dimension is $t - r$. All d_v columns can vary independently. Equation 26 follows. □

Corollary E.5 (Query-count condition). *If $m < t$, at least $(t - m)d_v$ value degrees of freedom remain unobserved. Full column rank requires $m \geq t$. Dependent attention rows can still prevent it.*

Let $a_\star = a_K(q_\star)$ and let Π_Q be the orthogonal projector onto $\mathrm{Null}(A_Q(K))$.

Proposition E.6 (Exposure at a new query). *For every $\rho > 0$,*

$$\sup_{\substack{A_Q(K)\Delta V = 0 \\ \|\Delta V\|_F \leq \rho}} \|\Delta V^\top a_\star\|_2 = \rho \|\Pi_Q a_\star\|_2. \tag{27}$$

Some value perturbation invisible to the observed readouts changes the new readout precisely when $\Pi_Q a_\star \neq 0$.

Proof. Feasible columns of ΔV lie in the null space, giving

$$\Delta V^\top a_\star = \Delta V^\top \Pi_Q a_\star.$$

Cauchy–Schwarz gives the upper bound. For nonzero $n = \Pi_Q a_\star$, choose unit vector $e \in \mathbb{R}^{d_v}$ and set

$$\Delta V = \rho \frac{n}{\|n\|_2} e^\top.$$

Perturbation ΔV attains $\rho\|n\|_2$. If the projection is zero, every feasible perturbation remains invisible at $q_\star$. □

Equation 27 shows that exposure is controlled by the component of $a_\star$ orthogonal to $\mathrm{Row}(A_Q(K))$. Query-vector distance alone does not determine this component. Query $q_\star$ adds no value constraint when its attention vector lies in the span of previously observed attention rows.

Statements here fix P and K. Some unseen queries may incur no error. Generalization bounds for a parameter-shared reconstruction network require additional assumptions.

E.5 Finite-Query Key Ambiguity

Let

$$\mathcal{U} = \mathrm{span}\{q_1, \ldots, q_m\} \subseteq \mathbb{R}^{d_h}.$$

Proposition E.7 (Unobserved key direction). *If $t \geq 2$ and $q_\star \notin \mathcal{U}$, distinct key matrices K and K' can give equal attention weights for every query in Q and different weights at $q_\star$.*

Proof. Choose $u \in \mathcal{U}^\perp$ with $u^\top q_\star \neq 0$. Choose $b \in \mathbb{R}^t$ that is not proportional to $\mathbf{1}_t$ and set

$$K' = K + bu^\top.$$

Every observed query satisfies

$$K' q_i = K q_i + b(u^\top q_i) = K q_i.$$

At $q_\star$, logit perturbation is $b(u^\top q_\star)/\sqrt{d_h}$, which varies across positions because b is nonconstant. Softmax weights change. Readout also changes whenever

$$V^\top [a_{K'}(q_\star) - a_K(q_\star)] \neq 0.$$

□

E.6 KEY-SHIFT INVARIANCE

Complete query coverage may still leave raw keys non-unique.

Proposition E.8 (Common key shift)**.** *For $c \in \mathbb{R}^{d_h}$, define $K_c = K + \mathbf{1}_t c^\top$. Every query satisfies*

$$a_{K_c}(q) = a_K(q), \qquad \mathcal{O}_{K_c,V}(q) = \mathcal{O}_{K,V}(q). \tag{28}$$

Proof. Shifted logits obey

$$\frac{K_c q}{\sqrt{d_h}} = \frac{Kq}{\sqrt{d_h}} + \mathbf{1}_t \frac{c^\top q}{\sqrt{d_h}}.$$

Softmax is invariant to a common additive score. Attention weights and readouts remain unchanged. □

Architectural constraints can reduce the equivalence class. Operator measurements remain well-defined even when raw tensor coordinates admit more than one representation.

E.7 DEPTH-ONLY PREDICTION FLOOR

Let U follow a specified conditional distribution over candidate branches at depth k. Assume $\mathbb{E}[\|q^{\mathrm{T}}_{U,k}\|_2^2 \mid P, k] < \infty$, set $Y = q^{\mathrm{T}}_{U,k}$, and let

$$\mu(P,k) = \mathbb{E}[Y \mid P, k].$$

For every branch-agnostic predictor $f(P,k)$,

$$\begin{aligned}\mathbb{E}[\|Y - f(P,k)\|_2^2 \mid P,k] &= \operatorname{tr}\operatorname{Var}(Y \mid P,k) \\ &\quad + \|\mu(P,k) - f(P,k)\|_2^2.\end{aligned} \tag{29}$$

Expanding $Y - f = (Y - \mu) + (\mu - f)$ gives the identity. Conditional expectation of the cross term is zero. Minimum occurs at $f = \mu$, leaving the covariance trace in Equation 8. Floor is positive whenever this trace is nonzero.

Equation 29 gives a prediction floor at fixed prefix and depth. Controlled evidence changes and readout-diversity measurements characterize the observed depth pattern.

F SPLIT-ATTENTION MERGE

F.1 TWO-SEGMENT MERGE

Let P denote prefix indices and let L_v denote local indices for candidate node v. Sets P and L_v are disjoint. Both kernels use the same query, score scale, positional coordinates, and visibility rule. Mask value m_i is zero for a visible position and $-\infty$ elsewhere. For segment $S \in \{P, L_v\}$, define

$$s_i = \frac{q^\top k_i}{\sqrt{d_h}} + m_i, \qquad Z_S = \sum_{i \in S} \exp(s_i),$$

$$N_S = \sum_{i \in S} \exp(s_i) v_i. \tag{30}$$

When $Z_S > 0$, set $\lambda_S = \log Z_S$ and $O_S = N_S / Z_S$. When $Z_S = 0$, set $\lambda_S = -\infty$ and $O_S = \mathbf{0}$. Formulas below assume $Z_P + Z_{L_v} > 0$.

Proposition F.1 (Two-segment merge)**.** *Attention over $P \sqcup L_v$ satisfies*

$$O = e^{\lambda_P - \lambda} O_P + e^{\lambda_{L_v} - \lambda} O_{L_v}, \qquad \lambda = \mathrm{LSE}(\lambda_P, \lambda_{L_v}). \tag{31}$$

Proof. Full normalizer is $Z = Z_P + Z_{L_v}$. Value-weighted numerator is $N_P + N_{L_v} = Z_P O_P + Z_{L_v} O_{L_v}$. Division by Z gives

$$O = \frac{Z_P}{Z} O_P + \frac{Z_{L_v}}{Z} O_{L_v}.$$

Substitution of $Z_S = e^{\lambda_S}$ yields Equation 31. □

Merged output equals full attention over $P \sqcup L_v$ in real arithmetic for every candidate query and attention head.

F.2 Stable Evaluation

For any segment with a visible position, set

$$c_S = \max_{\substack{i \in S \\ m_i = 0}} s_i, \qquad \widetilde{Z}_S = \sum_{\substack{i \in S \\ m_i = 0}} e^{s_i - c_S},$$
$$\widetilde{N}_S = \sum_{\substack{i \in S \\ m_i = 0}} e^{s_i - c_S} v_i, \qquad \lambda_S = c_S + \log \widetilde{Z}_S,$$
$$O_S = \frac{\widetilde{N}_S}{\widetilde{Z}_S}. \tag{32}$$

Let $\mathcal{A} = \{S \in \{P, L_v\} \mid Z_S > 0\}$ and $\lambda_{\max} = \max_{S \in \mathcal{A}} \lambda_S$. Stable merge calculations are

$$\lambda = \lambda_{\max} + \log \sum_{S \in \mathcal{A}} e^{\lambda_S - \lambda_{\max}},$$
$$O = \frac{\sum_{S \in \mathcal{A}} e^{\lambda_S - \lambda_{\max}} O_S}{\sum_{S \in \mathcal{A}} e^{\lambda_S - \lambda_{\max}}}. \tag{33}$$

F.3 Blockwise Accumulation

Corollary F.2 (Multiple segments)**.** *For pairwise-disjoint segments $S_1, \ldots, S_r$ with $\sum_{a=1}^{r} Z_{S_a} > 0$,*

$$\lambda = \mathrm{LSE}(\lambda_{S_1}, \ldots, \lambda_{S_r}), \qquad O = \sum_{a=1}^{r} e^{\lambda_{S_a} - \lambda} O_{S_a}. \tag{34}$$

Proof. Pair (O_S, λ_S) represents numerator and normalizer $(e^{\lambda_S} O_S, e^{\lambda_S})$. Segment accumulation adds both quantities. Associativity of addition gives Equation 34. □

Page blocks may be accumulated in any order in real arithmetic. Floating-point reduction order can introduce ordinary rounding differences. Shared prefix pages and path-local KV may use different tiling with identical masks and positional coordinates.

G Memory and Serving Measurements

G.1 Tensor Payload

Let t be verified-prefix length, m retained target hidden layers, d hidden width, D draft layers, n_{kv} KV heads, d_h head dimension, and b bytes per scalar. Retained target hidden state and contextual draft KV use

$$M_{\mathrm{tap}} = tbmd,$$
$$M_{\mathrm{ctxKV}} = tbD(2n_{\mathrm{kv}}d_h).$$

Reference hidden conditioning and eager fusion have payloads

$$M_{\mathrm{ref}} = M_{\mathrm{tap}} + M_{\mathrm{ctxKV}}, \tag{35}$$
$$M_{\mathrm{fused}} = tbd + M_{\mathrm{ctxKV}}. \tag{36}$$

Qwen3-8B BF16 uses $m = D = 5$, $d = 4096$, $n_{\mathrm{kv}} = 8$, $d_h = 128$, and $b = 2$. Per-token terms are

$$\frac{M_{\mathrm{tap}}}{t} = 2 \cdot 5 \cdot 4096 = 40 \text{ KiB}, \tag{37}$$
$$\frac{M_{\mathrm{ctxKV}}}{t} = 2 \cdot 5 \cdot (2 \cdot 8 \cdot 128) = 20 \text{ KiB}, \tag{38}$$
$$\frac{tbd}{t} = 2 \cdot 4096 = 8 \text{ KiB}. \tag{39}$$

At $t = 32{,}768$, analytical payloads are 1.875 GiB for the reference path with five hidden taps and contextual KV, and 0.875 GiB for eager fusion. Reported values include only the listed tensors. Root-anchor and active-tree state enter the next calculation.

Table 10: Sequence-linear drafter payload per prefix token.

Path	Hidden taps (KiB/token)	Contextual KV (KiB/token)	Total (KiB/token)
Reference hidden conditioning	40	20	60
Eager fused conditioning	8	20	28
OPSPEC sequence-linear prefix tensors	0	0	0

G.2 INCREMENTAL OPSPEC STATE

Learned OPSPEC state beyond target KV satisfies

$$M_{\mathrm{OpSpec}}^{\mathrm{learned}}(t, B) = M_{\mathrm{anchor}} + M_{\mathrm{tree}}(B).$$

Anchor size is constant in prefix length. Active-tree state grows with B. Fixed B gives constant learned incremental state in t.

Figure 3 reports measured incremental state of 0.032 GiB per request for OPSPEC and 0.91 GiB per request for JetSpec-N at 32K. Analytical eager-fusion payload is 0.875 GiB at $t = 32{,}768$. Measurements do not resolve the remaining 0.035 GiB by component.

Measured reduction is

$$1 - \frac{0.032}{0.91} = 96.5\%.$$

Target cache remains outside both incremental figures.

G.3 OPERATING POINTS

H100 serving measurements use 2K and 32K prompts. Offered concurrency C is fixed at 1 or 16. Maximum feasible batch at 32K is 10 for JetSpec-N and 12 for OPSPEC.

Table 11: Reported H100 throughput comparisons.

Operating point	Comparison	OPSPEC gain
$C = 1$	Equal offered concurrency	3.0%
$C = 16$	Equal offered concurrency	4.8%
Own maximum	Batch 12 versus batch 10	9.7%

Unfused native-prefix attention adds 5.6% to proposal latency. Removing hidden fusion and contextual KV projection recovers 4.2%. Fused split attention reduces proposal latency by 2.4% at 2K and 5.8% at 32K. Budget 128 gives highest throughput at moderate concurrency. Budget 256 gives largest offline accepted length.

G.4 DERIVED SERVING CONTRASTS

Measured state gap at 32K is $0.91 - 0.032 = 0.878$ GiB/request, corresponding to a state ratio of $0.91/0.032 = 28.4\times$. Maximum feasible batch gain is $(12 - 10)/10 = 20\%$. Fixed-concurrency throughput gain rises from 3.0% at $C = 1$ to 4.8% at $C = 16$, a difference of 1.8 percentage points. Own-maximum throughput gain is 4.9 points above the fixed-$C = 16$ result. Each row retains its listed operating point.

H EVALUATION SCOPE

Reported model evidence covers Qwen3-8B in non-thinking mode, listed datasets, and evaluated tree budgets. Coverage is limited to this target and drafter design.

Within each prefix-source comparison, checkpoint parameters, query states, tree construction, and target verification remain fixed. Full OPSPEC also trains the query and readout interface. Under

the reported oracle condition, target queries raise τ from 8.63 to 9.17 and measure query-related headroom.

Memory figures count incremental drafter state beyond target KV. OPSPEC still reads every verified-prefix token at selected layers. Serving results cover H100 hardware, 2K and 32K prompts, and listed operating points. Hardware, kernels, request mix, and batching policy affect throughput.

Split-attention identity requires matching score scale, positional encoding, visibility mask, and cached KV coordinates. Finite-precision reduction can change the final bits. Target verification determines committed output. Evidence Switching and task-level gains cover the listed evaluations.