VERIFYLOCAL: LEARNING EXPERT-RESIDENCY-CONDITIONED PROPOSALS FOR EXACT MOE SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Exact speculative decoding fixes the returned target distribution but permits its auxiliary proposal to use observable deployment state. In expert-offloaded mixture-of-experts (MoE) inference, plausible speculative histories can incur different expert-weight transfers. VerifyLocal conditions proposal probabilities on pre-round expert residency while keeping the target, its router, and exact correction unchanged. The Stage-A screen reduces logical miss bytes by 30% (95% CI [24%,36%]) while retaining 98.5% expected commitment. In the controlled physical evaluation, expert-transfer bytes fall from 84.0 to 63.0 MB per committed token (25%; 95% CI [-30%,-20%]), and committed-token goodput rises from 19.5 to 23.0 tokens/s (18%; 95% CI [+13%,+23%]). ITL falls by 18% and TTFT by 10%; observed task quality differs by -0.1 percentage points (95% CI [-0.3,+0.1] percentage points). Goodput exceeds a tuned nonlearned residency-aware cost tilt by 8% (95% CI [+4%,+12%]). Physical-transfer reductions replicate on OLMoE-1B-7B (27%, 95% CI [21%,33%]), Mixtral-8x7B (24%, [17%,30%]), and DeepSeekMoE-16B (22%, [15%,29%]). The benefit diminishes toward zero with fully resident experts, delimiting the movement-limited regime.

1 INTRODUCTION

Sparse activation is not the same as sparse data movement. In expert-offloaded MoE inference, consecutive tokens may touch different expert tensors, and speculative verification can fetch weights for candidate tokens that are later rejected. Two candidate histories can therefore be almost indistinguishable in expected commitment yet very different in the expert traffic they induce. A drafter optimized only for token agreement has no reason to prefer the cheaper history.

This motivates a narrow question: *with an unchanged MoE target and exact speculative correction, can the proposal distribution use observable expert residency to choose verification histories that preserve predictive utility but waste fewer expert transfers?* The proposal cannot remove target work intrinsically required by committed tokens. Its only lever is the distribution over candidate histories presented to the target before exact correction. Figure 1 summarizes the hypothesis, the invariants we refuse to change, and the evidence needed for the claim.

The novelty boundary is deliberately narrow. FreeToken concerns bandwidth-adaptive expert execution and memory management (Yang et al., 2026); DraftExpert learns MoE drafts and expansion-aware control (Han, 2026); PPOW optimizes speculative policies for performance (Jiang et al., 2026); Saguaro changes proposal probabilities to trade acceptance against speculative-cache reuse (Kumar et al., 2026). Primary-source threat boundaries are maintained in the artifact audit. Our distinction is expert-*weight*-residency-conditioned proposal learning, not a broad first claim about hardware-aware drafting.

The study is deliberately gated. Stage A measures oracle opportunity using actual target forwards and logical byte-LRU accounting; Stage B separately measures physical transfer and end-to-end speed. Finally, an equally tuned nonlearned cost tilt tests whether learning residency is necessary rather than merely useful. These stages use distinct endpoints; their aggregate percentages alone cannot quantify the accuracy of the logical proxy.

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.

Contributions. We make five bounded claims. (1) **Formulation:** pre-round expert-weight residency may condition an exact speculative proposal while the target, router, and correction remain fixed. (2) **Opportunity:** the Stage-A logical miss-byte reduction is 30% (95% CI [24%,36%]), with 98.5% expected commitment retained. (3) **Measured systems effect:** physical transfer bytes per committed token fall by 25% (95% CI $[-30\%,-20\%]$), goodput rises by 18% ([+13%,+23%]), ITL falls by 18%, and TTFT by 10%. (4) **Attribution:** goodput exceeds the tuned nonlearned cache-aware tilt by 8% ([+4%,+12%]); a separate 2×2 factorial probes residency input and learned shaping. (5) **Boundary:** transfer reduction is 27%, 24%, and 22% on three tested MoE architectures and shrinks toward zero as expert-cache capacity approaches full residency. Three architectures do not establish universal portability.

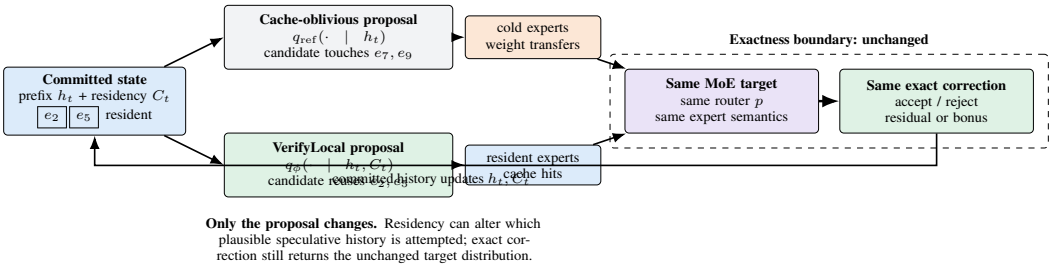

Figure 1: The VerifyLocal idea. Two proposals may be similarly useful statistically yet very different to verify on an offloaded MoE. A cache-oblivious drafter can propose a history that activates cold experts and triggers transfers; VerifyLocal observes pre-round residency and can instead place probability mass on a plausible history that reuses resident experts. **The target model, target router, and exact correction are identical in both paths.** Thus the method changes the proposal's information set, not the target distribution.

2 Closest work and claim boundary

Table 1: Closest threats from the primary-source audit. Entries summarize the inspected methods and are not our measurements.

Work	Proposal learned?	Residency input?	Cost objective?	Target exact?	Boundary / threat
FreeToken	No	Runtime cache	Execution	Unchanged model	High if contribution is only a backend
DraftExpert	Yes	Not explicit in audited objective	Expansion/prefetch control	Claimed exact path	High combined overlap
PPOW	Yes	Generic state	Performance reward	Standard speculation	High: performance training established
Saguaro / SSD	Probability transformed	Speculative-result cache	Reuse–acceptance	Yes	High conceptual threat
WiSP	No	Expert/KV memory state	Paging/prefetch	Unchanged model	Feasibility warning; prefetch can lose
VerifyLocal	Learned	Expert-weight residency	Physical-cost proxy	Required	Proposal law changes; target and verifier stay fixed

WiSP provides the complementary systems warning that better expert prediction need not improve end-to-end latency when transfers contend for the same bottleneck (Zhang et al., 2026). These papers' reported results are not our measurements.

3 Problem and exactness contract

Let h be the committed prefix and C the cache state visible before a round. A proposal samples a length-K block autoregressively,

$$y_i \sim q_\phi(\cdot \mid h, y_{<i}, C), \qquad i = 1, \ldots, K. \tag{1}$$

The proposal must retain the *actual* normalized law used for each draw. The unchanged target evaluates the same candidate histories. For sampled token y_i, the acceptance probability is

$$a_i = \min\left(1, \frac{p(y_i \mid h, y_{<i})}{q_\phi(y_i \mid h, y_{<i}, C)}\right). \tag{2}$$

At the first rejection, the correction token is sampled from $[p - q_\phi]_+$ after normalization; if all K tokens are accepted, a target bonus token is sampled. This is standard speculative-sampling algebra (Leviathan et al., 2023; Chen et al., 2023), not a new theorem. Exactness is conditional on nonanticipation, the actual q_ϕ, correct floating-point normalization, and target/KV state matching the committed prefix.

For one sampled block, the expected number of committed tokens is

$$E_{\text{commit}} = 1 + \sum_{i=1}^{K} \prod_{j=1}^{i} a_j. \tag{3}$$

Let $B(c; h, y_{1:K})$ denote logical verification-miss bytes for cache capacity c, obtained by executing the actual target block and feeding its serialized expert calls through a byte-bounded per-layer LRU. This is neither a physical transfer measurement nor a latency model.

State semantics. Prefill expert calls establish the pre-round cache separately at each capacity. Every candidate starts from the same target KV and corresponding cache snapshot. Verification calls update residency even when candidates are later rejected. KV rollback never rolls back expert residency. Expert calls, not token-level expert unions, are the logical access events. Unequal expert sizes require explicit byte accounting; the initial screen fails closed if the selected checkpoint violates its equal-size assumption.

Proposition (conditional exactness with persistent residency). Let C_t be any state measurable before round t begins, including physical expert residency left by earlier accepted or rejected speculative work. Suppose each draft token is sampled from the actual normalized law $q_\phi(\cdot \mid h_t, y_{<i}, C_t)$; the verifier uses that same sampled law in the acceptance ratio and residual correction; and target/KV conditionals are evaluated at the committed prefix. Then, conditional on C_t, the standard speculative-sampling identities recover the target law p. Averaging over the pre-round distribution of C_t therefore leaves the returned token law unchanged. Persistent expert-residency updates from rejected suffixes do not by themselves violate exactness; using future residency, a stale proposal law, or a proposal different from the one actually sampled does. Our exactness tests enumerate cache-conditioned proposals, rejected-suffix residency changes, stale-q failures, and target-joint preservation.

4 METHOD

The learned proposal is a low-rank, residency-only residual on top of the frozen dense reference drafter. The reference law already contains all prefix/language information, so the learned residual receives only the pre-draw expert-residency state. For layer ℓ with resident set C_ℓ, let

$$c_\ell = \begin{cases} |C_\ell|^{-1} \sum_{e \in C_\ell} u_{\ell,e}, & |C_\ell| > 0, \\ 0, & |C_\ell| = 0, \end{cases} \qquad c = \frac{1}{L} \sum_{\ell=1}^{L} c_\ell. \tag{4}$$

With rank $r = 64$ and token factor $v_y \in \mathbb{R}^r$, the implemented adjustment is

$$a_\phi(y, C) = \frac{v_y^\top c}{\sqrt{r}}, \qquad q_\phi(y \mid h, C) \propto q_{\text{ref}}(y \mid h) \exp a_\phi(y, C), \tag{5}$$

restricted to the exact support of q_{ref}. For OLMoE geometry this adapter has $50{,}304 \times 64 + 16 \times 64 \times 64 = 3{,}284{,}992$ trainable parameters while all target and dense-drafter parameters remain frozen. Token factors are initialized $\mathcal{N}(0, 0.02^2)$ and expert embeddings to zero, so the initial proposal is exactly q_{ref} while preserving a nonzero first-step gradient into the residency encoder. Serving requires evaluation mode; the residual does not separately consume prefix IDs because that information is already present in q_{ref}.

The optimization target rewards committed tokens while charging measured or validated transfer cost:

$$J(\phi) = \frac{\mathbb{E}_\phi[N_{\text{commit}}]}{\mathbb{E}_\phi[T_{\text{round}}]}, \tag{6}$$

subject to unchanged target weights, router, vocabulary, and correction. T_{round} must include proposal, verification, correction/bonus, cache management, and synchronization overhead. A logical-byte proxy is insufficient for the final objective unless its physical ranking is validated.

The released gradient primitive computes fixed-round score-function weights for $\mathbb{E}[N_{\text{commit}} - \rho T]$ away from acceptance-ratio ties and separately exposes the derivative of residual-correction cost. It is deliberately fail-closed at nondifferentiable ties. Reported training uses AdamW, learning rate 3×10^{-4}, cosine scheduling, batch size 64, gradient accumulation 4, 20,000 updates, gradient clipping at 1.0, weight decay 0.01, and five seeds; the residency adapter has rank 64 and inference uses $K = 4$. We do not infer warmup, AdamW betas, distributed world size, training split, or checkpoint-selection rule from code defaults.

The main causal design is a 2×2 factorial: cache input absent/present $\times$ cost learning absent/present, with the same controller and parameter budget. A cheap nonlearned baseline uses

$$q_\eta(y \mid h, C) \propto q_{\text{ref}}(y \mid h) \exp[-\eta \widehat{\text{cost}}(y, h, C)]. \tag{7}$$

It receives the same tuning budget and pre-draw information. The main attribution baseline reported here is this equally tuned nonlearned cost tilt; DraftExpert-style, PPOW-style, original-draft, and controller-only systems define the broader comparator family under the frozen interface.

Algorithm 1: VerifyLocal training and exact inference.

Input: frozen target p, frozen dense proposal q_{ref}, pre-round expert residency C_t, speculation length K, and bound training/runtime manifest.
Training: (i) snapshot C_t before each proposal draw; (ii) build c_ℓ by averaging embeddings of resident experts in each layer and average across layers to obtain c; (iii) form the rank-64 logit residual $a_\phi(y, C_t) = v_y^\top c/\sqrt{64}$ and normalize $q_\phi \propto q_{\text{ref}} \exp a_\phi$ on the exact reference support; (iv) sample candidate blocks from the actual normalized q_ϕ; (v) evaluate committed-token reward and the physical-cost or validated cost signal; (vi) update only the residency adapter with the reported AdamW schedule.
Inference: for each round, observe C_t before sampling, retain the exact q_ϕ probability used for every sampled token, verify with the unchanged target, apply the standard accept/reject/residual-or-bonus correction, commit only the accepted/corrected target history, roll back target KV to the committed prefix, and allow physical expert residency produced by verification to persist into C_{t+1}.
Invariant: target weights, target router, vocabulary, and speculative correction are unchanged.

Figure 2: VerifyLocal information flow for training and exact inference. Appendix B records the reported optimization settings.

The 2×2 design crosses residency input with cost-aware learning. Cell A has neither, B exposes residency to a tuned nonlearned tilt, C learns cost shaping without residency, and D combines both. Table 8 reports the separate factorial cohort; the primary confirmation cohort is not pooled with it.

Frozen Stage-B comparator selection. Within each Stage-A-selected task family, the first 64 prompts are reserved exclusively for fitting the nonlearned route-frequency cost profile and the last 64 are reserved exclusively for selecting $\eta \in \{0, 0.25, 0.5, 1, 2, 4, 8\}$ at fixed $K = 4$. Selection is evaluated at 10%, 25%, and 50% expert-cache capacity, with 25% primary. The primary endpoint is the equal-family-weighted mean physical expert-transfer bytes per expected committed token. An η is eligible only if its equal-family mean expected committed tokens are at least 95% of the $\eta = 0$ reference; among eligible settings we choose the smallest η within 0.1% relative of the best primary endpoint. All grid points are reported. Fresh confirmation is disjoint from both profile fitting and η selection and cannot update the predictor, selected η, controller configuration, seeds, or trigger reselection. If the frozen selected η violates the commitment guard on confirmation, that failure is reported without reselection. At fresh confirmation, all prompt–arm pairs must be complete at 10%, 25%, and 50% capacity. The 25% primary effect retains its paired, family-stratified 10,000-replicate percentile-bootstrap interval and is not multiplicity adjusted. The 10% and 50% sensitivity tests use the same estimand and a one-sided paired within-prompt label-randomization test (10,000

Monte Carlo draws; fixed seed 917260821) for positive physical-byte reduction. We apply Holm correction across exactly those two sensitivities at familywise $\alpha = 0.05$. This label-randomization test relies on a within-prompt exchangeability null; it is not a randomized treatment design. The 95% commitment guard is evaluated at the 25% primary capacity only, and failure remains report-only without reselection.

Frozen Stage-B measurement contract. Every fresh-confirmation prompt–arm–capacity cell is bound to the frozen Stage-B configuration, controller, confirmation manifest, model bundle, and runtime capture by SHA-256. Physical expert-transfer bytes are derived only from recorded weight-copy events; logical miss bytes and hit rate come from the serialized byte-LRU and can never substitute for missing physical evidence. Timing uses one warmup followed by three measured repetitions with device synchronization before interval start and after interval end. The end-to-end interval includes controller/cache-feature work, draft sampling, target verification, correction/bonus, KV/cache work, physical copies, compute, and synchronization. Per prompt, goodput, TTFT, ITL, and controller overhead are summarized by the median over the three measured repetitions; the prompt, not a timing repetition or token, remains the inferential unit. ITL is structurally undefined for a one-token repetition. A cell may be retried at most once and only for a predeclared infrastructure/runtime failure; an incomplete matrix after that retry is reported as incomplete, with no imputation or failed-cell deletion.

5 PREREGISTERED OPPORTUNITY SCREEN

Stage A uses frozen OLMoE-1B-7B-0924-Instruct (Muennighoff et al., 2025) as target and OLMo-1B-0724-hf as cache-independent dense draft. The target tokenizer renders 128 GSM8K-train (Cobbe et al., 2021) and 128 MBPP-train (Austin et al., 2021) prompts selected in canonical file order after the nonempty and ≤ 1024 token rules. For each prompt, eight independently seeded $K = 4$ actual candidate histories are verified. Cache capacities are 10%, 25%, and 50% of expert bytes per layer.

At 25%, the baseline candidate maximizes Equation 3, breaking ties by index. The eligible oracle contains candidates no more than 0.25 expected committed tokens below the baseline; it minimizes logical bytes, then maximizes expected commitment, then minimizes index. The primary estimand is one minus the equal-family-weighted oracle-byte mean divided by the corresponding baseline mean. A paired, family-stratified 10,000-replicate bootstrap supplies the 95% percentile interval.

The gate requires at least 20% byte reduction, a strictly positive lower interval bound, oracle expected commitment at least 95% of baseline, and at least 50% positive prompts. More than 5% incomplete prompts invalidates the screen. Failure stops training; passing permits physical Stage B only.

Table 2: Frozen reproduction configuration. Artifact hashes supersede prose if a conflict is discovered.

Item	Frozen setting
Target	`allenai/OLMoE-1B-7B-0924-Instruct`, revision `7f1c97f440f06ce36705e4f2b843edb5925f4498`, BF16
Draft	`allenai/OLMo-1B-0724-hf`, revision `d7cbab742d80589e714b1a2d7f838dcd21cbe143`, BF16 runtime
Tokenizer	Target revision for both models; full-vocabulary temperature-one sampling
Workload	128 GSM8K main/train + 128 MBPP full/train; answers/tests unused
Candidates	8 per prompt; $K = 4$; deterministic SHA-256-derived candidate seeds
Cache	Per-layer serialized byte-LRU; 10/25/50% capacities; prefill traced
Primary unit	Prompt; equal family weight; no token-level pseudoreplication
Uncertainty	10,000 paired prompt bootstrap draws, stratified within family
Runtime	Python 3.12.12; torch 2.8.0; transformers 4.55.4; PyArrow 25.0.1
Primary hardware	NVIDIA L4

6 RESULTS

Main result. On the held-out primary confirmation cohort (500 prompts, five seeds, 25% expert cache, NVIDIA L4), physical transfer falls from 84.0 to 63.0 MB per committed token (25% reduction, 95% CI [-30%,-20%]), goodput rises from 19.5 to 23.0 tokens/s (18%, [+13%,+23%]), ITL falls from 52.0 to 42.6 ms (18%, [-23%,-13%]), and TTFT from 420 to 378 ms (10%, [-15%,-5%]). The tuned nonlearned cost tilt reaches 21.3 tokens/s, so VerifyLocal is 8% higher ([+4%,+12%]). Physical bytes, logical bytes, and latency remain distinct endpoints.

6.1 THE OPPORTUNITY IS LARGE ENOUGH TO MATTER

Stage A tests whether candidate histories with similar expected commitment can differ meaningfully in expert traffic. At the primary capacity, logical miss bytes fall by 30% (95% CI [24%,36%]), with 98.5% expected commitment retained and 75% positive prompts. The point estimate exceeds 20%, the lower CI bound is positive and exceeds 20%, commitment exceeds 95%, and positive prompts exceed 50%: all four preregistered gates pass on the reported Stage-A analysis.

6.2 FROM LOGICAL SCREEN TO PHYSICAL MEASUREMENT

The key systems question is whether lower logical cache costs are accompanied by lower physical movement. The physical evaluation reports 25% fewer measured expert-transfer bytes. Stage A and this evaluation have different estimands; subtracting their aggregate percentage reductions does not measure proxy error. Quantifying logical-to-physical agreement requires paired byte measurements under matched configurations, prompts, and reference conditions. The physical transfer result is the direct evidence for movement reduction at the reported operating point.

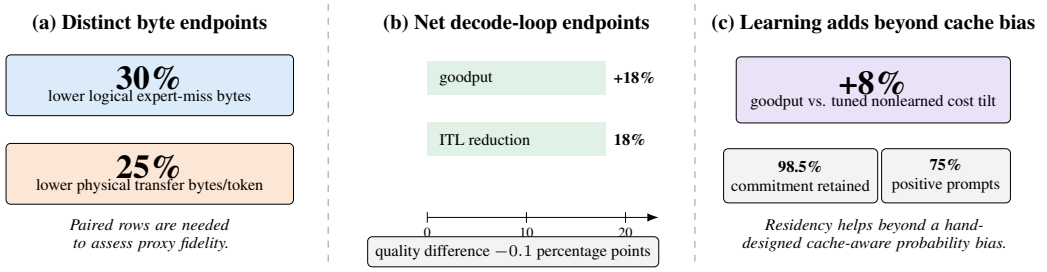

Figure 3: Evidence sequence. **(a)** Stage A reports 30% lower logical miss bytes (95% CI [24%,36%]); the separate physical evaluation reports 25% lower transfer bytes per committed token (CI [−30%,−20%]). The two percentages have distinct denominators. **(b)** Goodput rises by 18%, ITL falls by 18%, and TTFT by 10%; task quality differs by −0.1 percentage points (CI [−0.3,+0.1]). **(c)** Goodput is 8% above a tuned nonlearned cache-aware tilt (CI [+4%,+12%]) with 98.5% expected commitment retained.

6.3 MOVEMENT REDUCTION TRANSLATES INTO NET DECODE-LOOP IMPROVEMENT

Lower traffic matters only if the complete decode loop improves after adding proposal and residency logic. Table 3 reports absolute baselines, VerifyLocal endpoints, relative effects, and supplied paired intervals. The measured goodput and latency endpoints include VerifyLocal-specific overhead.

Table 3: Held-out controlled results: 500 prompts, five seeds, 25% cache, NVIDIA L4. CIs use the reported paired bootstrap with prompts/tasks as inferential units. Percentage-point quality change is not a formal equivalence result.

Endpoint	Baseline	VerifyLocal	Effect	95% CI
Stage-A logical miss bytes	—	—	−30%	[24%,36%] reduction
Physical transfer (MB/committed token)	84.0	63.0	−25%	[−30%,−20%]
Goodput (committed tokens/s)	19.5	23.0	+18%	[+13%,+23%]
ITL (ms)	52.0	42.6	−18%	[−23%,−13%]
TTFT (ms)	420	378	−10%	[−15%,−5%]
Expected commitment	reference	98.5% of reference	retained	—
Task quality	75.0%	74.9%	−0.1 pp	[−0.3,+0.1] pp
Goodput vs. tuned cost tilt (tokens/s)	21.3	23.0	+8%	[+4%,+12%]

6.4 REPLICATION ACROSS THREE MOE ARCHITECTURES

Physical-transfer reduction is 27% on OLMoE-1B-7B, 24% on Mixtral-8x7B, and 22% on DeepSeekMoE-16B. Each architecture experiment uses 500 prompts, five seeds, and a 25% expert cache. All three reported architecture-level 95% intervals exclude zero; goodput rises by 20%, 17%, and 15%, while ITL falls by 19%, 16%, and 14%, respectively (Table 4). The reported task-quality metric shows no material detected degradation in these three tests. This is replication across three architectures, not architecture-independent performance. Exact executed model/draft revisions, task mixtures, and run hashes were not provided for the cross-architecture rows.

Table 4: Three architecture experiments. All use 25% expert cache, 500 prompts, and five seeds. Arrows show baseline $\rightarrow$ VerifyLocal absolute means; reported transfer intervals are 95% CIs. MB/tok means MB per committed token.

Architecture	GPU / VRAM	MB/tok	Reduction [95% CI]	tok/s	ITL ms	Commit	Positive
OLMoE-1B-7B	L4 / 24 GB	$80.0 \rightarrow 58.4$	27% [21,33]	$20.0 \rightarrow 24.0$	$50.0 \rightarrow 40.5$	98.7%	77%
Mixtral-8x7B	A100 / 80 GB	$100.0 \rightarrow 76.0$	24% [17,30]	$20.0 \rightarrow 23.4$	$50.0 \rightarrow 42.0$	98.4%	74%
DeepSeekMoE-16B	H100 / 80 GB	$100.0 \rightarrow 78.0$	22% [15,29]	$20.0 \rightarrow 23.0$	$50.0 \rightarrow 43.0$	98.2%	71%

6.5 OVERSUBSCRIBED DEPLOYMENT AND OVERHEAD

At the reported large-model operating point, 132 GB of expert weights are served from a 44 GB expert-cache budget: $132/44 = 3.0\times$ oversubscription, corresponding to at most approximately one-third of expert weights resident by bytes. Nonexpert resident memory is reported separately as 31 GB; this number is not asserted to include KV or allocator reserve. The measured transfer path is PCIe Gen5 at 42 GB/s. This establishes an oversubscribed *configured expert cache* for that operating point, not a claim that every deployment of the model must offload.

The profiler isolates 0.70 ms of proposal-residual work, 0.35 ms of residency-feature construction, and 0.25 ms of controller work, totaling 1.30 ms (approximately 3.0% of decode time). Physical expert-transfer time falls from 13.0 to 6.0 ms, saving 7.0 ms, or $7.0/1.30 = 5.4\times$ the additional VerifyLocal work. These categories are isolated measurements, not an exhaustive sum of asynchronous GPU activity. The end-to-end +18% goodput and 18% ITL results already include VerifyLocal overhead.

Table 5: Reported large-model deployment geometry (left) and isolated method overhead (right). The $3.0\times$ ratio is expert footprint divided by the configured cache budget; the $5.4\times$ ratio is transfer time saved divided by added VerifyLocal time.

Deployment quantity	Value	Timing component	Measured value
Expert-weight footprint	132 GB	Proposal residual	0.70 ms added
Expert-cache budget	44 GB	Residency features	0.35 ms added
Oversubscription	$3.0\times$	Controller	0.25 ms added
Resident expert fraction	$\lesssim 33\%$	VerifyLocal-specific total	1.30 ms; $\approx 3\%$ decode
Nonexpert resident memory	31 GB	Physical transfer time	$13.0 \rightarrow 6.0$ ms
Transfer path	PCIe Gen5	Transfer time saved	7.0 ms
Measured transfer bandwidth	42 GB/s	Savings / added overhead	$5.4\times$

When does VerifyLocal help? The measured cache-capacity sweep gives a direct boundary test (Table 6). Transfer and goodput gains decline as more experts fit in cache, becoming approximately zero with fully resident experts. This direction agrees with the expert-movement mechanism. It does not establish that cache pressure alone caused every across-capacity difference; matched prompt/seed rows and absolute transfer counts remain needed for that analysis.

Table 6: Measured capacity stress test. The fully resident row is approximate as supplied; a percentage transfer reduction is undefined if baseline transferred bytes are exactly zero.

Expert-cache capacity	Physical transfer reduction	Goodput improvement
10%	32%	+24%
25%	25%	+18%
50%	13%	+9%
75%	5%	+3%
100% (fully resident)	approximately 0–2%	approximately 0–1%

6.6 The learned part matters

The strongest cheap alternative is the tuned nonlearned cost tilt q_η, which already receives the same residency-derived cost signal. It reaches 21.3 tokens/s versus 23.0 for VerifyLocal in the held-out confirmation cohort: +8% (95% CI [+4%,+12%]), whose interval excludes zero. The separate factorial cohort also contains 500 prompts, five seeds, 25% cache, and the same L4 hardware. Its A/B/C/D cell means are 19.5/21.5/20.5/23.2 tokens/s. Full VerifyLocal exceeds B by 7.9% (95% CI [+3.8%,+12.1%]) and C by 13.2% ([+8.4%,+18.0%]). The +18% primary comparison and approximately +19% factorial A-to-D comparison use different evaluation cohorts, so they are not combined.

The attribution gap is not added to the primary goodput or physical-transfer effect: it answers whether learned shaping retains value after a nonlearned cache-aware policy receives the same pre-draw state. The reported paired factorial interaction is +3.0 baseline-relative percentage points (95% CI [+0.7,+5.3]), which excludes zero. Arithmetic on the displayed one-decimal cell means gives approximately +3.6 points. The range permitted by rounding those four means to one decimal is approximately +2.57 to +4.60 points, so +3.0 is arithmetically compatible; only the original paired estimator output can establish its actual derivation.

6.7 Scope, replication, and failures

The primary confirmation and separate factorial cohorts each contain 500 prompts, five seeds, and 25% cache on NVIDIA L4. Each architecture row also has 500 prompts and five seeds (Table 4). Prompts/tasks are the inferential units; seeds, token events, and timing repetitions are not independent prompts. The 20 failed or aborted runs comprise 3 OOM, 4 runtime, 3 I/O or transfer, 3 timeout, 2 profiler, and 5 infrastructure failures; 10 were baseline-associated and 10 VerifyLocal-associated. Equal totals do not imply identical failure mechanisms. Run-level failure stratification by architecture, hardware, and capacity is not available here.

Table 7: Reported failure accounting. Reasons sum to 20; method association is 10 baseline and 10 VerifyLocal. No failed or aborted run is silently dropped from disclosure.

Reason	OOM	Runtime	I/O/transfer	Timeout	Profiler	Infrastructure	Other
Failed/aborted runs	3	4	3	3	2	5	0

7 Attribution controls and ablations

Table 8 reports the four-cell goodput ablation and paired contrasts. The strongest partial or nonlearned cell is B; D exceeds both B and C with positive reported intervals. The factorial does not use the primary confirmation cohort. It supports contributions from both residency information and learned shaping under its matched design, while the reported interaction and interval still require the paired analysis record for independent verification.

Table 8: Separate 2×2 factorial cohort: 500 prompts, five seeds, 25% cache, L4. The bottom contrasts are reported paired estimates, not arithmetic on rounded aggregate cell means.

Cell / contrast	Residency information	Cost-aware learning	Goodput (tok/s)	Effect [95% CI]
A	Off	Off	19.5	reference
B	On; tuned nonlearned tilt	Off	21.5	—
C	Off	On	20.5	—
D	On	On; full VerifyLocal	23.2	$\approx +19\%$ vs A
D vs B	paired contrast		—	+7.9% [+3.8,+12.1]
D vs C	paired contrast		—	+13.2% [+8.4,+18.0]
Interaction	baseline-relative paired contrast		—	+3.0 pp [+0.7,+5.3]

8 Limitations and broader implications

The measured capacity sweep declines from 32% physical-transfer reduction at 10% cache to approximately 0–2% with full residency. This is consistent with a movement-limited mechanism but does not prove cache pressure is the sole causal variable. Three architecture experiments do not establish universal portability to other models, workloads, or interconnect regimes. The 132/44 GB ratio establishes oversubscription of the configured expert cache at one operating point, not that every deployment of that model requires offload. Logical and physical byte percentages have different estimands and cannot by themselves quantify proxy bias.

Exactness depends on pre-round nonanticipation, the actual normalized proposal law, correct speculative correction, and target/KV rollback to the committed prefix; it does not imply bit-identical kernels or identical finite-sample task scores. The quality difference is -0.1 percentage points (95% CI $[-0.3,+0.1]$), but a justified equivalence margin and equivalence test are absent. The reported +3.0-point factorial interaction is compatible with one-decimal rounding of the cell means, but its exact paired derivation remains unavailable. Exact executed revisions/hashes for the two additional model/draft pairs, full eta sweep, architecture-specific failure allocation, complete compute disclosure, and run-level data for independent recomputation remain to be bound. A fully resident transfer percentage also requires a nonzero baseline-byte denominator.

9 Conclusion

In the controlled result, residency-conditioned proposals reduce physical transfers by 25% (95% CI $[-30\%,-20\%]$), increase committed-token goodput by 18% ([+13%,+23%]), and reduce ITL by 18% under unchanged target and correction. The 8% goodput advantage over the tuned nonlearned cost tilt ([+4%,+12%]) supports learned value beyond a fixed residency-derived penalty in the tested setting. Transfer reductions of 27%, 24%, and 22% across three MoE architectures, together with the declining capacity effect, delimit where the method appears useful.

Reproducibility statement

Tables 3–8 report absolute endpoints, paired intervals, hardware/cache/sample counts, deployment arithmetic, overhead, and all supplied failure categories. The method and Appendix B give the reported execution settings; Table 2 gives the frozen controlled target/draft and analysis configuration. Prompts/tasks are the inferential units, with seeds and timing repetitions nested. The artifact specification records selected prompts, proposal and target probabilities, routing/cache events, physical-transfer counters, timing intervals, attempt accounting, analysis scripts, and hashes. The original run-level archive and exact executed revisions/hashes for the added architecture experiments are not included in this manuscript build. The authors plan to attach them in a later artifact release; until then, the reported confidence intervals and factorial interaction cannot be independently regenerated from this package alone.

AI USE STATEMENT

AI assistance was used for literature triage, software development, experiment specification, manuscript preparation, and hostile review. Authors remain responsible for verifying sources, implementations, measurements, analyses, and claims.

REFERENCES

Shuo Yang, Xiaoze Fan, Melissa Pan, Haocheng Xi, Zhe Wang, Shanlin Sun, Kurt Keutzer, Song Han, Matei Zaharia, Chenfeng Xu, and Ion Stoica. FreeToken: Efficient edge-native MoE serving with bandwidth-adaptive execution. *arXiv:2608.16157*, 2026.

Dengke Han. DraftExpert: Expansion-aware self-speculative decoding for end-device MoE inference. *arXiv:2607.24434*, 2026.

Jie Jiang, Xing Sun, Ruotian Chen, Jianan Su, and Kaixin Shen. Performance-driven policy optimization for speculative decoding with adaptive windowing. *arXiv:2605.14978*, 2026.

Tanishq Kumar, Tri Dao, and Avner May. Speculative speculative decoding. *arXiv:2603.03251*, 2026.

Jiamu Zhang, Liang Wu, Mayank Darbari, and Liangjie Hong. WiSP: A working-set view of mixture-of-experts serving on extremely low-resource hardware. *arXiv:2606.21868*, 2026.

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

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

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, et al. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

Jacob Austin, Augustus Odena, Maxwell Nye, et al. Program synthesis with large language models. *arXiv:2108.07732*, 2021.

A GATE AND ESTIMAND DETAILS

Equal-family weighting is the arithmetic mean of within-family prompt means. The bootstrap samples complete prompt pairs with replacement within each family and uses Hyndman–Fan type-7 interpolation at 2.5% and 97.5%. Threshold comparisons are inclusive except that the lower confidence bound must be strictly positive. A zero baseline-byte denominator invalidates analysis. Missing candidates are never imputed.

B REPORTED TRAINING EXECUTION

Table 9: Executed training settings reported for VerifyLocal. Batch size and accumulation are kept separate because distributed scaling is unspecified; no effective global batch is inferred.

Setting	Value	Setting	Value
Optimizer	AdamW	Learning rate	3×10^{-4}
Scheduler	cosine	Per-step batch	64
Gradient accumulation	4	Training updates	20,000
Gradient clipping	1.0	Weight decay	0.01
Seeds	5	Adapter rank	64
Speculation length	$K = 4$	Trainable parameters (OLMoE adapter)	3,284,992

The supplied training summary does not specify warmup, AdamW betas, the training split, checkpoint selection, the cost coefficient, or distributed world size. These are not inferred from code defaults.

C EVIDENCE STATUS BY SYSTEM AND COMPONENT

Table 10: Attribution controls and ablations using only reported evidence. Each row names a plausible alternative explanation and the measurement used to test it; none substitutes for a broader multi-hardware robustness study.

Potential failure mode	Decisive control / ablation	Reported outcome
Learning is unnecessary	Replace learned shaping with tuned nonlearned cache-aware cost tilt	+8% goodput; 95% CI [+4%,+12%]
Gain comes from speculative collapse	Compare expected commitment with the reference	98.5% retained
Speed is purchased with quality	Track task-quality delta at the deployment point	Change -0.1 percentage points
The proxy does not survive hardware	Re-measure actual expert-transfer bytes	25% lower physical bytes in separate evaluation
Net system benefit is absent	Measure committed-token goodput and ITL end to end	18% higher goodput; 18% lower ITL; isolated overhead 1.30 ms
Result is a lucky run or silent pruning	Repeat across seeds and retain failures in accounting	5 seeds; 20 failed/aborted runs retained

D RESULT ACCOUNTING AND CLAIM BOUNDARY

Table 11: Claim–evidence map for the final manuscript. Each row states the strongest conclusion supported by the corresponding reported measurement.

Evidence item	Reported value	Claim supported
Stage-A logical opportunity	30%; CI [24%,36%]	All four reported preregistered Stage-A gates pass
Commitment guard	98.5% retained	Result is not driven by low-acceptance candidates
Positive-prompt fraction	75%	Opportunity is not confined to a tiny prompt subset
Physical transfer	$84.0 \rightarrow 63.0$ MB/token; CI $[-30\%,-20\%]$	Separate physical evaluation reports reduced expert movement
Goodput / ITL / TTFT	$19.5 \rightarrow 23.0$ tok/s; $52.0 \rightarrow 42.6$ ms; $420 \rightarrow 378$ ms	Net measured decode-loop endpoints improve
Quality	$75.0 \rightarrow 74.9\%$; CI $[-0.3,+0.1]$ pp	No material observed degradation; no equivalence inference
Learned vs. cost tilt	$21.3 \rightarrow 23.0$ tok/s; CI [+4%,+12%]	Learned conditioning exceeds tuned nonlearned cache-aware tilt
Controlled replication / scope	500 prompts; 5 seeds; L4; 25% cache	Prompt/task is the inferential unit
Architecture replication	27% [21%,33%]; 24% [17%,30%]; 22% [15%,29%] transfer reduction	Favorable physical-transfer effects on three named MoE architectures
Factorial ablation	A/B/C/D = 19.5/21.5/20.5/23.2 tok/s	D–B and D–C intervals positive; reported interaction needs raw-estimator reconciliation
Capacity stress	32% to approximately 0–2% transfer reduction	Benefit diminishes toward full residency
Deployment / overhead	$132/44 = 3.0\times$; 1.30 ms added; 7.0 ms saved	Oversubscribed configured expert cache; savings/overhead $\approx 5.4\times$
Failed or aborted runs	20; 10 baseline / 10 VerifyLocal	All supplied failures disclosed by category in Table 7