KNOW BEFORE YOU ROUTE: DUAL-SIGNAL EXPERT PREFETCHING WITH CONFORMAL COVERAGE GUARANTEES FOR SPARSE MOE LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Mixture-of-Experts (MoE) LLMs achieve high capacity with sparse activation, but under GPU memory constraints the full expert pool may not fit on device, forcing expert-weight transfers from host memory onto the decode critical path. We introduce **CEPA**, a retraining-free dual-signal expert prefetching framework. An early velocity-corrected cross-layer proxy initiates one likely-expert transfer, while a later signal evaluated on the non-extrapolated layer-boundary residual constructs a calibrated expert set. The boundary residual remains a proxy for the post-attention router input, yet corrected lower-tail conformal calibration provides layerwise marginal whole-active-set coverage when calibration and test score–set pairs are exchangeable. CEPA speculates only on transfer timing: exact routing and expert computation are preserved by fetching every missing active expert on demand. Across four sparse MoE LLMs spanning 14.3B–35B total parameters under a 12 GB GPU memory budget, CEPA reduces decode latency by up to $1.37\times$ and retains a $1.32\times$ speedup against a matched CUDA-Graph baseline.

1 INTRODUCTION

Sparse Mixture-of-Experts (MoE) language models increase capacity while activating only a small fraction of their parameters per token (Shazeer et al., 2017; Jiang et al., 2024; Qwen Team, 2024). When the active state and KV cache fit on the GPU but the expert pool does not, routed expert weights must cross PCIe before each MoE block executes (Figure 1). Recent systems reduce this cost through CPU–GPU orchestration, activation-aware caching, and learned prefetching (Kamahori et al., 2024; Xue et al., 2024; Song et al., 2024; Fang et al., 2025; Zhu et al., 2025). We target exact GPU expert execution in this memory-constrained, latency-oriented regime and evaluate batch sizes through eight.

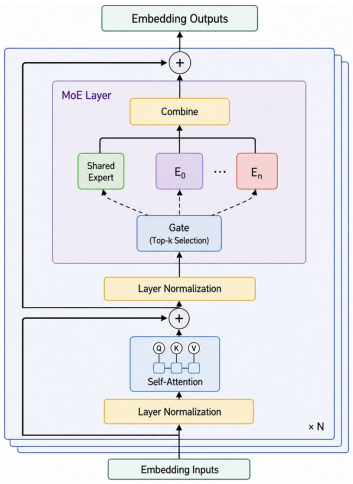


Figure 1: Sparse MoE offloading: selected experts move from host memory over PCIe.

The central difficulty is timing. The exact router for a layer acts only after that layer's attention or token-mixing block, so just-in-time (JIT) fetching starts late and leaves transfer latency on the critical path. CEPA separates early initiation from later correction. Signal E_A fires after the preceding layer's attention block and uses a velocity-corrected cross-layer proxy to launch one likely expert; signal E_B fires at the target-layer boundary and uses the non-extrapolated boundary residual to form a variable-size set. Conformalizing the earlier proxy requires 16.8–18.5 experts at the nominal 90% level, versus 4.79 for E_B, motivating this asymmetric point-plus-set design.

E_B calibrates the minimum boundary-signal score among the true active experts. A corrected lower-tail conformal order statistic provides layerwise marginal whole-active-set coverage $1-\alpha$ when calibration and test score–set pairs are exchangeable. The guarantee concerns the abstract prediction set, not residency by a transfer deadline; because our evaluation pools dependent autoregressive tokens, measured coverage is reported as an empirical diagnostic rather than finite-sample validation of the

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.

theorem. CEPA changes only transfer timing: any missing active expert is fetched on demand before use, and unused prefetched experts are never executed. Unlike speculative expert execution (Madan et al., 2026), prediction errors therefore affect latency and bandwidth but not model outputs.

Across four sparse MoE LLMs, CEPA reduces decode latency by $1.10\times$–$1.37\times$ and retains a $1.32\times$ gain over a matched CUDA-Graph JIT baseline. Memory-budget, hardware, batching, and 4K-context controls preserve positive gains, and CEPA and JIT produce bit-identical logits and tokens.

Contributions. CEPA contributes **(1)** a retraining-free dual-signal runtime that separates early transfer initiation from set-valued correction while preserving exact MoE execution; **(2)** a corrected lower-tail conformal construction with exchangeability-conditional layerwise coverage and minimality within its nested threshold family (Thm. 4); and **(3)** a matched systems evaluation with CUDA-Graph, memory-budget, hardware, batching, long-context, exactness, and profiling controls.

2 Related Work

Prior work on MoE inference spans CPU–GPU orchestration, activation-aware caching and scheduling, learned predictors, and speculative or compressed execution. These systems make different choices about where experts execute, which models they support, whether predictors are trained, and whether the target is single-request latency or multi-request throughput. CEPA focuses on exact GPU expert execution under host-weight offloading and separates early transfer initiation from calibrated active-set coverage.

2.1 Conformal Prediction Background

Conformal prediction constructs prediction sets with finite-sample marginal coverage under exchangeability (Vovk et al., 2005). Conformal Risk Control (CRC) extends this principle to monotone losses through a corrected empirical-risk threshold (Angelopoulos et al., 2024).

CEPA applies CRC independently at each layer to the minimum boundary-signal score among the routed experts. The resulting threshold set contains the entire top-K active set with marginal probability at least $1-\alpha$ under exchangeability. This statement requires neither router retraining nor a uniformly accurate boundary proxy; Section 3.3 defines the score and threshold precisely.

2.2 MoE Offloading and Expert Prediction

Fiddler executes selected non-resident experts on the CPU to avoid moving their weights, whereas CEPA retains GPU expert execution and moves weights earlier (Kamahori et al., 2024). MoE-Infinity and FineMoE emphasize activation-aware caching and fine-grained offloading (Xue et al., 2024; Yu et al., 2025b). DuoServe-MoE and LayerScope target dual-phase or multi-batch scheduling regimes (Zhang et al., 2025; Yu et al., 2025a). Their native systems use different models, hardware, memory policies, or serving objectives, so we do not present their published end-to-end numbers as matched comparisons. Appendix O reports only an accuracy-matched runtime *sensitivity replay*. Aggregate accuracy does not determine the layers, cache states, lead times, or temporal correlation of errors, so the replay is not evidence of superiority over either native system.

Learned and architecture-modifying predictors address expert selection more directly. Pre-gated MoE adds a pre-gating function and changes the model execution path (Hwang et al., 2024). Fate uses adjacent-layer signals with confidence thresholds, but relies partly on over-fetching, cache assistance, and expert quantization (Fang et al., 2025). Pre-Attention Expert Prediction trains lightweight pre-attention routers, whereas ProMoE uses intermediate activations to drive proactive caching (Zhu et al., 2025; Song et al., 2024).

Speculative and compression-based methods reduce transfer cost by changing execution or representation. Madan et al. (2026) execute prefetched experts speculatively, so wrong predictions can alter computation.

Quantization reduces transfer and residency costs but changes the numerical execution. CEPA is orthogonal to quantization: for a fixed deployed model, it preserves that model's routed computation and uses prediction only to move communication earlier.

2.3 Positioning of CEPA

Many prediction-based approaches still trade prefetch lead time against expert-selection fidelity. Early signals provide more overlap but are less accurate, while accurate routing signals arrive too

late for bandwidth-efficient prefetching. CEPA addresses this coupling with two signals: a velocity-corrected point predictor for immediate transfer initiation and a layer-boundary-residual conformal set for marginal coverage, without requiring large persistent caches, auxiliary retraining, architectural modification, speculative execution, or aggressive quantization.

3 METHOD

Overview. CEPA assigns earliness and coverage to different signals. **Signal** E_A fires after $\mathrm{Attn}_{\ell-1}$ but before $\mathrm{MoE}_{\ell-1}$, using a velocity-corrected proxy to launch one early transfer. Conformalizing this signal is valid but empirically bandwidth-inefficient (§3.2.1).

Signal E_B fires at the $(\ell{-}1){\to}\ell$ boundary and uses the non-extrapolated boundary residual to construct a CRC set with marginal $1{-}\alpha$ coverage. Its scoring variant is selected before calibration (§3.3.1). CEPA unions the point prediction and threshold set, inheriting E_B's guarantee under exchangeability while retaining E_A's longer communication window. The notation is defined below.

3.1 PROBLEM FORMULATION AND NOTATION

Model and Residual Stream. Let $\mathcal{M}$ be an L-layer language model with E *routed* experts per MoE layer and top-K routing ($K \ll E$); always-active shared experts are modeled separately. Write $[E] = \{1, \ldots, E\}$, let t index decode tokens, and let H be the residual width. Each layer $\ell \in \{0, \ldots, L{-}1\}$ consists of a model-native token-mixing block (standard self-attention or linear attention, denoted Attn_ℓ) followed by an MoE block (MoE_ℓ). We track the residual stream at two times:

- $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^H$: the *layer-boundary* residual entering layer ℓ (output of layer $\ell{-}1$'s complete MoE block; token embedding at $\ell{=}0$).
- $\mathbf{h}_t^{(\ell,\mathrm{mid})} \in \mathbb{R}^H$: the *mid-layer* residual at layer ℓ (output of Attn_ℓ plus its residual connection; input to MoE_ℓ). Thus $\mathbf{h}_t^{(\ell+1)}$ is unavailable until MoE_ℓ completes.

Router and Expert Activation. We use row-vector notation. Each layer ℓ has router weights $\mathbf{W}_r^{(\ell)} \in \mathbb{R}^{H\times E}$ and a pre-router normalization $\mathrm{LN}_r^{(\ell)}$. Given the mid-layer residual, the router produces logits and model-native scores:

$$\mathbf{z}_t^{(\ell)} = \mathrm{LN}_r^{(\ell)}\big(\mathbf{h}_t^{(\ell,\mathrm{mid})}\big)\mathbf{W}_r^{(\ell)}, \qquad \mathbf{p}_t^{(\ell)} = \rho_r^{(\ell)}\big(\mathbf{z}_t^{(\ell)}\big) \in [0,1]^E, \tag{1}$$

where $\rho_r^{(\ell)}$ is the model-native router score transformation (e.g., softmax); softmax-specific bounds below are labeled explicitly. Let $p_{t,(k)}^{(\ell)}$ denote the k-th largest score. The top-K active set is $\mathcal{A}_t^{(\ell)} = \mathrm{TopK}(\mathbf{p}_t^{(\ell)}, K) \subseteq [E]$, and the full MoE update is:

$$\mathbf{h}_t^{(\ell+1)} = \mathbf{h}_t^{(\ell,\mathrm{mid})} + s^{(\ell)}\big(\mathbf{h}_t^{(\ell,\mathrm{mid})}\big) + \sum_{e\in\mathcal{A}_t^{(\ell)}} g_{t,e}^{(\ell)}\, f_e^{(\ell)}\big(\mathbf{h}_t^{(\ell,\mathrm{mid})}\big), \tag{2}$$

where $f_e^{(\ell)} : \mathbb{R}^H \to \mathbb{R}^H$ is routed expert e, $g_{t,e}^{(\ell)}$ is its exact model-native mixing weight, and $s^{(\ell)}$ aggregates any always-active shared-expert branches ($s^{(\ell)} \equiv 0$ when none exist). Shared branches remain resident and are not prediction targets. The active set $\mathcal{A}_t^{(\ell)}$ is determined by $\mathbf{h}_t^{(\ell,\mathrm{mid})}$, which does not exist until Attn_ℓ completes; this is CEPA's fundamental timing constraint.

The Expert Prefetching Problem. When not all E expert tensors can reside on device, let $\mathcal{R}_t^{(\ell)} \subseteq [E]$ be the experts already resident when routing finishes. A JIT runtime must transfer $M_t^{(\ell)} = |\mathcal{A}_t^{(\ell)} \setminus \mathcal{R}_t^{(\ell)}|$ missing experts after the routing decision, placing up to $M_t^{(\ell)}\tau_{\mathrm{fetch}}$ on the critical path, where τ_{fetch} is the serialized transfer time of one routed expert, when those transfers cannot be overlapped. A prefetch set $\hat{\mathcal{S}}_t^{(\ell)} \subseteq [E]$ identified before MoE_ℓ instead permits communication to overlap with earlier computation. Set containment is a statistical coverage property; complete latency hiding additionally requires that the covered, non-resident experts finish transferring before they are consumed.

Let $\mathcal{D}_t^{(\ell)}$ be the experts resident at consumption time, including cache hits and completed transfers. The conformal event $\mathcal{A}_t^{(\ell)} \subseteq \hat{\mathcal{S}}_t^{(\ell)}$ does not imply the deadline event $\mathcal{A}_t^{(\ell)} \subseteq \mathcal{D}_t^{(\ell)}$ under finite slots or bandwidth; deadline coverage and transfer waste therefore require separate measurement.

Definition 1 (Valid Prefetch Set). *A prefetch set $\hat{\mathcal{S}}_t^{(\ell)}$ is* valid at level $1-\alpha$ *(miscoverage budget $\alpha \in (0,1)$, e.g. α=0.1) if $\mathbb{P}\big[\mathcal{A}_t^{(\ell)} \subseteq \hat{\mathcal{S}}_t^{(\ell)}\big] \geq 1-\alpha$. We use expected set size as the efficiency objective; on every covered example, the number of excess experts is exactly $|\hat{\mathcal{S}}_t^{(\ell)}| - K$. We seek* $\min\ \mathbb{E}[|\hat{\mathcal{S}}_t^{(\ell)}|]$ *subject to the coverage constraint.*

Earlier predictions widen the overlap window but require coarser approximations of $\mathbf{h}_t^{(\ell,\mathrm{mid})}$, inflating set size; later ones are more accurate but narrow the window. CEPA resolves this by assigning *overlap maximisation* and *coverage validity* to separate signals.

3.2 Signal E_A: Velocity-Corrected Early Point Prediction

Cross-Layer Proxy and Velocity Extrapolation. Signal E_A fires when $\mathbf{h}_t^{(\ell-1,\mathrm{mid})}$ becomes available (after $\mathrm{Attn}_{\ell-1}$, before $\mathrm{MoE}_{\ell-1}$), opening an overlap window covering $\mathrm{MoE}_{\ell-1}$ and the full Attn_ℓ. Let $\phi_r^{(\ell)}(\mathbf{h}) \triangleq \mathrm{LN}_r^{(\ell)}(\mathbf{h})\mathbf{W}_r^{(\ell)}$ be the composed router map. The true logit $\mathbf{z}_t^{(\ell)} = \phi_r^{(\ell)}(\mathbf{h}_t^{(\ell,\mathrm{mid})})$ is unavailable at E_A firing time, so we evaluate the target-layer router on the two most recent mid-layer states, $\mathbf{z}_t^{(\ell|j)} = \phi_r^{(\ell)}(\mathbf{h}_t^{(j,\mathrm{mid})})$. Their difference is a discrete logit velocity, yielding

$$\tilde{\mathbf{z}}_{t,\varepsilon}^{(\ell)} = \mathbf{z}_t^{(\ell|\ell-1)} + \varepsilon\Big(\mathbf{z}_t^{(\ell|\ell-1)} - \mathbf{z}_t^{(\ell|\ell-2)}\Big), \qquad \varepsilon \geq 0, \tag{3}$$

and $\hat{e}_t^{A,(\ell)} = \arg\max_e \tilde{z}_{t,\varepsilon^{*(\ell)},e}^{(\ell)}$. We select $\varepsilon^{*(\ell)}$ on a finite grid using data disjoint from the CRC calibration split. When the $\ell-2$ state is unavailable, CEPA uses the zero-order fallback $\varepsilon^{*(\ell)} = 0$.

Proposition 2 (Diagnostic Velocity Approximation Bounds). *Fix the target router $\phi_r^{(\ell)}$ and define the interpolated logit trajectory $G(s) = \phi_r^{(\ell)}(\mathbf{h}_t^{(s,\mathrm{mid})})$. Assume G is twice continuously differentiable on $[\ell-2, \ell]$ with $\|G'(s)\|_2 \leq B_1$ and $\|G''(s)\|_2 \leq B_2$. The logit approximation errors satisfy:*

$$\|\tilde{\mathbf{z}}_{t,0}^{(\ell)} - \mathbf{z}_t^{(\ell)}\|_2 \leq B_1, \tag{4}$$

$$\|\tilde{\mathbf{z}}_{t,1}^{(\ell)} - \mathbf{z}_t^{(\ell)}\|_2 \leq B_2. \tag{5}$$

Thus, when $B_2 < B_1$, the unit-velocity proxy has the tighter worst-case upper bound. This comparison concerns only $\varepsilon = 0$ and 1; CEPA selects ε from a grid. It neither establishes $B_2 < B_1$ empirically nor implies a higher top-K hit rate.

Proof sketch. The fundamental theorem of calculus bounds the zero-order difference by B_1. The unit-velocity error is the second finite difference $G(\ell) - 2G(\ell-1) + G(\ell-2)$; integral Taylor remainders bound it by B_2. Full proof: Appendix A. □

3.2.1 Why CEPA Does Not Conformalize the Early Proxy

CRC applied to the early proxy would retain its marginal coverage guarantee under exchangeability, but validity alone does not imply a useful prefetch set size. For softmax routers, a logit perturbation $\Delta\mathbf{z}$ obeys $|\log \tilde{p}_{t,e} - \log p_{t,e}| \leq 2\|\Delta\mathbf{z}\|_\infty$, so proxy error can move the lower-tail active-expert scores that determine the threshold. The bound does *not* establish a stochastic ordering between early and late signals.

The design choice is empirical: conformalized early proxies require 16.8–18.5 experts on average at 90% coverage, versus 4.79 for E_B (Table 3). CEPA therefore uses E_A only for one advance transfer and assigns set-valued coverage to E_B. Appendix B formalizes the sensitivity bound.

3.3 Signal E_B: Conformal Coverage Sets

Signal E_B fires at the $(\ell-1)\to\ell$ boundary when $\mathbf{h}_t^{(\ell)}$ becomes available. It forms the boundary-signal logits and scores $\bar{\mathbf{z}}_t^{B,(\ell)} = \phi_r^{(\ell)}(\mathbf{h}_t^{(\ell)})$ and $\bar{\mathbf{p}}_t^{B,(\ell)} = \rho_r^{(\ell)}(\bar{\mathbf{z}}_t^{B,(\ell)})$, which approximate the true router outputs available only after Attn_ℓ. If $\phi_r^{(\ell)}$ is σ_ℓ-Lipschitz on the relevant residual states, its logit error is at most $\sigma_\ell\|\mathbf{h}_t^{(\ell,\mathrm{mid})} - \mathbf{h}_t^{(\ell)}\|_2$.

This deterministic bound is explanatory only. CRC is calibrated directly on the pairs $(\bar{\mathbf{p}}_t^{B,(\ell)}, \mathcal{A}_t^{(\ell)})$ and its validity does not require the boundary proxy to approximate the true logits uniformly.

Definition 3 (γ-Coverage Set and CRC Calibration). *Define the threshold set as $\mathcal{S}_t^{(\ell)}(\gamma) = \{e \in [E] : \bar{p}_{t,e}^{B,(\ell)} \geq \gamma\}$ and the calibration score as $m_t^{B,(\ell)} = \min_{e\in\mathcal{A}_t^{(\ell)}} \bar{p}_{t,e}^{B,(\ell)}$. The miscoverage indicator is*

$\mathcal{L}_t^{(\ell)}(\gamma) \triangleq \mathbf{1}[\mathcal{A}_t^{(\ell)} \not\subseteq \mathcal{S}_t^{(\ell)}(\gamma)] = \mathbf{1}[m_t^{B,(\ell)} < \gamma]$. *The CRC threshold is the largest threshold whose finite-sample corrected calibration risk does not exceed* α*:*

$$\gamma^{*(\ell)} = \sup\left(\{0\} \cup \left\{\gamma \in (0,1] : \frac{1}{N_{\text{cal}}+1}\Big[\sum_{t=1}^{N_{\text{cal}}} \mathcal{L}_t^{(\ell)}(\gamma) + 1\Big] \le \alpha\right\}\right) \equiv \widehat{Q}_\alpha^{\text{CRC}}\big(\{m_t^{B,(\ell)}\}_{t=1}^{N_{\text{cal}}}\big), \tag{6}$$

where $\widehat{Q}_\alpha^{\text{CRC}}$ *is a corrected* lower-tail *empirical quantile (not a* $(1-\alpha)$*-quantile), because miscoverage is the event* $m_t^{B,(\ell)} < \gamma$*.*

Theorem 4 (Marginal Coverage and Threshold-Family Minimality)**.** *If the* N_{cal} *calibration pairs and one test pair* $(\bar{\mathbf{p}}^{B,(\ell)}, \mathcal{A}^{(\ell)})$ *are exchangeable, then* $\mathbb{P}[\mathcal{A}_{N_{\text{cal}}+1}^{(\ell)} \subseteq \mathcal{S}_{N_{\text{cal}}+1}^{(\ell)}(\gamma^{*(\ell)})] \ge 1-\alpha$*. Moreover,* $\gamma^{*(\ell)}$ *yields the pointwise smallest set* within the nested γ-threshold family *subject to the same corrected empirical-risk constraint; no optimality is claimed over other set predictors or latency-aware schedules.*

Proof. The strict-loss convention makes miscoverage equivalent to the test minimum score falling below a corrected lower-tail calibration order statistic. Exchangeability bounds that rank event by α; monotonicity in γ gives threshold-family optimality. Full proof: Appendix H. Assumptions and layerwise composition are discussed in Appendices I–L. □

3.3.1 Selecting the Boundary-Signal Variant

Unlike E_A, the boundary signal has no universally optimal extrapolation rule: a perturbation may raise or lower active-expert scores, so softmax sensitivity alone cannot determine the CRC threshold or set size. We select the E_B scoring rule on a model-selection split and then calibrate $\gamma^{*(\ell)}$ on a disjoint conformal split; this separation is required to retain the stated finite-sample guarantee after variant selection. Table 3 reports 9.10 experts for a vanilla pre-normalization boundary proxy, 5.18 after jointly changing the normalization and enabling velocity, and 4.79 for the selected target-router normalization with velocity disabled. Hence the $9.10 \to 5.18$ comparison is confounded and is not evidence that velocity helps. The controlled post-normalization comparison is $5.18 \to 4.79$ when velocity is removed; CEPA's E_B implementation uses one boundary state and no blending. Appendix E records the two-sided sensitivity bound.

3.4 Dual-Signal Fusion: Resolving the Asymmetry

Structural asymmetry. E_A fires after $\text{Attn}_{\ell-1}$ but before $\text{MoE}_{\ell-1}$, evaluates the target router on earlier mid-layer states, and returns one expert. Its layerwise velocity coefficient is selected for rank-1 accuracy and transfer lead time; it is not used for set coverage. E_B fires later, after $\text{MoE}_{\ell-1}$ but before Attn_ℓ, evaluates the target router on the non-extrapolated boundary residual $\mathbf{h}_t^{(\ell)}$, and returns a variable-size threshold set. Only E_B carries the conditional conformal statement. The signals are therefore complementary rather than redundant: E_A opens the longer DMA window, while E_B controls prediction-set risk. CEPA takes their union:

$$\hat{\mathcal{S}}_t^{(\ell)} = \underbrace{\{\hat{e}_t^{A,(\ell)}\}}_{\text{E}_A} \cup \underbrace{\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})}_{\text{E}_B}. \tag{7}$$

Proposition 5 (Coverage and Size of the Hybrid Set)**.** *Under the exchangeability assumptions of Theorem 4,* $\hat{\mathcal{S}}_t^{(\ell)}$ *is valid at level* $1-\alpha$ *(Definition 1). For any joint distribution of the two signals, its expected size satisfies*

$$\begin{aligned}\mathbb{E}\big[|\hat{\mathcal{S}}_t^{(\ell)}|\big] &= \mathbb{E}\big[|\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})|\big] \\ &\quad + \mathbb{P}\big[\hat{e}_t^{A,(\ell)} \notin \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\big].\end{aligned} \tag{8}$$

Proof sketch. Coverage follows because $\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)}) \subseteq \hat{\mathcal{S}}_t^{(\ell)}$. The union adds one element exactly when $\hat{e}_t^{A,(\ell)}$ is absent from the threshold set; linearity of expectation yields (8). See Appendix C. □

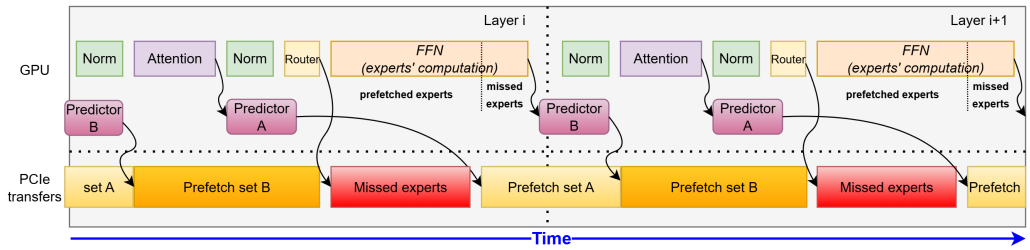


Figure 2: CEPA runtime timeline. Signals E_A and E_B prefetch next-layer experts early, overlapping PCIe transfers with current-layer computation while preserving exact MoE execution.

Inference Protocol. At the completion of $\mathrm{Attn}_{\ell-1}$, E_A evaluates (3) and launches one DMA transfer for $\hat{e}_t^{A,(\ell)}$ while $\mathrm{MoE}_{\ell-1}$ executes. Once $\mathrm{MoE}_{\ell-1}$ completes, E_B evaluates $\phi_r^{(\ell)}(\mathbf{h}_t^{(\ell)})$, forms $\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})$, and requests its additional non-resident experts. After the exact router for layer ℓ executes, every active expert still absent is fetched on demand before use. Thus predictions change only communication timing, never expert selection or numerical computation. Offline selection and calibration require forward passes over their respective held-out splits; online scoring costs $O(HE)$ and set filtering costs $O(E)$ per target layer.

3.5 Runtime Co-Design for Cross-Layer Prefetching

The previous subsections specify how CEPA produces predictive signals; here we describe how the runtime uses them. Routing accuracy alone is insufficient: latency improves only if the required experts are resident on the GPU when the MoE block consumes them. CEPA therefore couples cross-layer prefetching with a runtime that maximizes communication–computation overlap while preserving memory-efficiency.

3.5.1 Baseline and Design Objective

The conventional JIT baseline transfers missing expert weights after routing, overlapping only intra-layer transfer with shared or resident expert computation. This opens the communication window too late: PCIe remains idle during attention because expert identities are unknown, and GPU bubbles appear once expert execution begins because transfer is slower than expert computation. CEPA addresses both inefficiencies by starting transfers for layer $\ell{+}1$ while layer ℓ is still running, without changing exact computation at consumption time; Appendix P provides aligned JIT and CEPA timelines.

3.5.2 Two-Stage Cross-Layer Prefetch Schedule

CEPA uses a two-stage schedule matching E_A and E_B. At the end of Attn_ℓ, the early signal launches exactly one transfer for $\hat{e}_t^{A,(\ell+1)}$. Once the $(\ell{\to}\ell{+}1)$ boundary residual $\mathbf{h}_t^{(\ell+1)}$ is available, E_B constructs the calibrated set and queues each member not already resident or in flight. Thus E_A uses the longer but less accurate window, while E_B corrects and completes the request set from the more reliable boundary state.

Figure 2 illustrates this overlap. While the GPU finishes layer ℓ, CEPA issues staged PCIe transfers for layer $\ell{+}1$. Experts requested by E_A or E_B may therefore be resident when layer $\ell{+}1$ reaches its MoE block. After exact routing, any missing active expert falls back to on-demand transfer, which can overlap with resident or shared-expert computation when dependencies permit. Unused prefetched experts are never executed. CEPA therefore speculates only on transfer timing and remains numerically identical to the JIT execution for every prediction outcome.

Quantifying the Prefetch Budget. Let $W_A^{(\ell+1)}$ be the measured time from E_A firing at the end of Attn_ℓ until the target MoE begins, and let $W_B^{(\ell+1)}$ be the shorter time from E_B firing at the layer boundary until that same deadline. Writing $T_{\mathrm{rem}}^{(\ell)}$ for the remaining current-layer compute, these windows satisfy

$$W_A^{(\ell+1)} = T_{\mathrm{rem}}^{(\ell)} + W_B^{(\ell+1)}, \qquad W_B^{(\ell+1)} \le W_A^{(\ell+1)}. \tag{9}$$

For equal-size experts with serialized transfer time T_{transfer}, the single E_A transfer is fully hidden whenever $T_{\mathrm{transfer}} \le W_A^{(\ell+1)}$. If $M_B = |\mathcal{S}_t^{(\ell+1)} \setminus \mathcal{R}_t^{(\ell+1)}|$ E_B experts remain to be transferred,

Table 1: Sparse MoE models. "Shared?" indicates always-active shared-expert computation in addition to the routed pool; E and top-K refer only to routed experts.

Model	Total Params.	Active Params.	Shared?	Top-K	# Experts	Hidden Size	Expert Intermediate Size
Qwen1.5-MoE	14.3B	2.7B	Yes	4	60	2048	1408
Qwen3.5-35B-A3B	35B	3B	Yes	8	256	2048	512
Gemma 4 26B-A4B	25.2B	3.8B	Yes	8	128	2816	704
GPT-OSS-20B	21B	3.6B	No	4	32	2880	2880

at most $\min\{M_B, \lfloor W_B^{(\ell+1)}/T_{\text{transfer}} \rfloor\}$ serialized transfers can finish entirely within the boundary window. This accounting is a deadline bound, not a coverage statement: Theorem 4 controls which experts appear in the predicted set, whereas realized overlap also depends on PCIe contention, residency, expert size, and scheduling.

3.5.3 Parallel Runtime Organization

On the CPU, a main thread drives the forward pass while a background thread handles prefetch orchestration, candidate filtering, and transfer issuance. On the GPU, a prefetch stream carries expert host-to-device transfers while the primary stream executes normalization, token mixing, routing, and expert kernels, maximizing communication–computation overlap with minimal synchronization.

3.5.4 Residency and Overhead Control

CEPA exploits short-range temporal locality through a layer-local residency policy: experts consumed for token t in layer ℓ are retained and reused for token $t+1$ in the same layer if selected again, while unused speculative experts are evicted first, reducing redundant PCIe traffic when routing is persistent across consecutive decode tokens.

The deployment fixes a per-layer prefetch-slot capacity B_{pf}. Candidates are ordered by signal score and issued only while both a slot and the measured deadline budget in (9) remain. Excess set members stay in the abstract prediction set but are not promised resident by the MoE deadline; if exact routing selects them, they are fetched on demand. Thus truncation leaves model outputs unchanged and does not alter the statistical coverage of the *prediction set*, but it does remove any corresponding coverage claim for the truncated resident prefix. Peak speculative residency is deterministically bounded by B_{pf}. In particular, $\mathbb{P}[\mathcal{A}_t^{(\ell)} \subseteq \hat{\mathcal{S}}_t^{(\ell)}] \geq 1-\alpha$ does not imply $\mathbb{P}[\mathcal{A}_t^{(\ell)} \subseteq \mathcal{D}_t^{(\ell)}] \geq 1-\alpha$.

Orchestration overhead is kept small by using two CPU threads and two GPU streams: the main thread and compute stream execute the forward pass, while the background thread and prefetch stream handle candidate filtering and expert transfers. Together with normalization reuse and minimal synchronization, these runtime mechanisms make CEPA a co-design rather than a predictor-only add-on: they jointly optimize signal timing, transfer dispatch, host–device overlap, miss recovery, inter-token residency, and memory footprint so that the predictive sets from §3.2–§3.4 yield end-to-end decoding speedups under strict autoregressive latency constraints.

4 Experiments

We evaluate CEPA on four sparse MoE language models spanning different router designs, expert counts, top-K values, and hidden dimensions: Qwen1.5-MoE (Qwen Team, 2024), Qwen3.5-MoE (Qwen Team, 2026), Gemma 4 26B A4B (Gemma Team, 2026), and GPT-OSS-20B (OpenAI, 2025). Table 1 summarizes their architectures.

4.1 Hardware and Memory-Constrained Offloading Setting

The main four-model comparison runs on a single NVIDIA RTX A6000 GPU with usable memory capped at 12 GB via PyTorch memory controls, creating a controlled offloading regime where the expert pool cannot remain resident. Attention, routing, shared experts, and other non-expert components remain on GPU; routed expert weights are transferred from pinned host memory. CEPA changes only transfer timing.

The cap models a common low-memory regime but does not reproduce every property of a native 12 GB device. Appendix O.5 therefore adds 8/12/16 GB budgets, a separate RTX 2000 Ada, and a controlled GPU-frequency experiment.

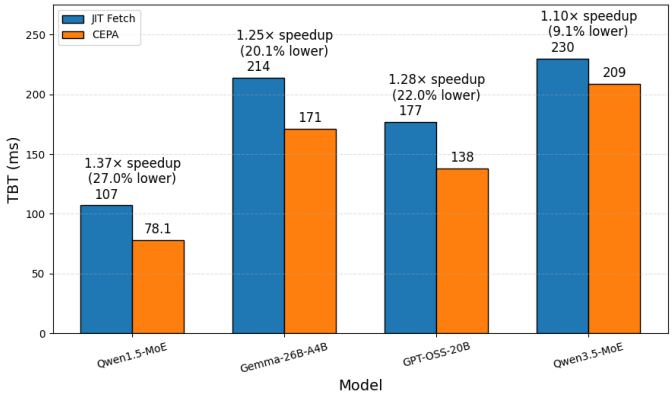


Figure 3: Mean decode latency under the matched 12 GB JIT and CEPA runtime. CEPA changes transfer timing but executes the same routed experts.

4.2 Implementation

We implement both modes in the same custom PyTorch MoE path. The **JIT baseline** transfers only the exact top-K experts after routing; **CEPA** advances predicted transfers and fetches every remaining active expert on demand. Both use identical routing, residency, kernels, and memory budgets, isolating transfer timing.

In the matched CUDA-Graph experiment, both modes capture normalization, attention with a static KV allocation, and the shared-expert MLP. Dynamic routed-expert copies remain outside the graph. CEPA and JIT produce bit-identical logits and tokens over the evaluation set (Appendix O.1).

4.3 End-to-End Decode Latency and Execution Ablation

Figure 3 compares CEPA against the JIT baseline. CEPA reduces decode latency on all models: Qwen1.5-MoE from 107 ms to 78.1 ms (1.37×), Gemma-26B-A4B from 214 ms to 171 ms (1.25×), GPT-OSS-20B from 177 ms to 138 ms (1.28×), and Qwen3.5-MoE from 230 ms to 209 ms (1.10×). The gains arise from overlapping host-to-device DMA with non-expert GPU compute.

Qwen3.5-MoE has the smallest gain because its smaller experts shorten both transfer and compute, leaving less exposed latency per layer. Appendix P reports Nsight timelines and per-layer PCIe utilization.

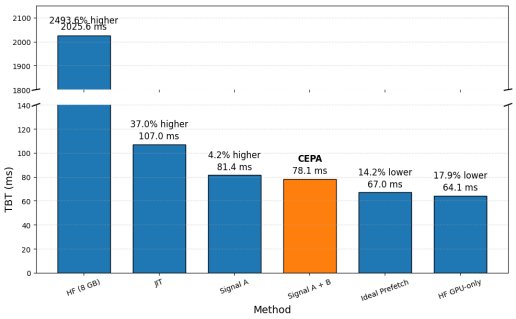


Figure 4: Qwen1.5-MoE execution ablation.

Execution ablation. Figure 4 separates the two signals. E_A reduces 107.0 to 81.4 ms and therefore supplies $25.6/28.9 = 88.6\%$ of the total JIT-to-CEPA reduction. Adding E_B reduces 81.4 to 78.1 ms, an additional 3.3 ms or 4.1% relative to E_A alone. Thus the main latency mechanism is early transfer initiation; the measured incremental systems value of E_B is smaller and accompanies its risk-control interpretation. We did not run matched fixed-size top-k or ordinary-threshold boundary policies, so these results do not establish lower latency than those simpler alternatives.

Deployment robustness. CEPA retains $1.32\times$ speedup with CUDA Graphs, 21.7–26.4% gains across 8–16 GB budgets, 11–23% gains on RTX 2000 Ada, and positive speedup through batch 8 and 4K context. Appendix O reports the configurations, percentiles, overheads, and sensitivity replays.

Runtime interpretation. With matched CUDA Graphs, JIT and CEPA require 91.2 and 69.1 ms/token, respectively. E_B scoring and thresholding costs 0.117 ms per predicted layer, or 2.69 ms across 23 layers, and is included in every CEPA latency. Prediction-set coverage is neverthe-

Table 2: Prediction quality at α=0.1. C_{union} is pooled-token whole-set containment of the union, not a confidence interval or an E_B coverage estimate.

			E_A			E_B		*CEPA*				
Model	K	E	r_1	r_K	Adv.	r_1	r_K	r_1	r_K	C_{union}	$\lvert\hat{\mathcal{S}}\rvert$	Speedup
Qwen1.5	4	60	0.940	0.414	✓	0.995	0.718	0.940	0.719	0.900	4.83	1.37×
Qwen3.5	8	256	0.958	0.395	✓	1.000	0.882	0.958	0.882	0.905	8.07	1.10×
Gemma	8	128	0.978	0.491	✓	1.000	0.946	0.978	0.946	0.898	7.36	1.25×
GPT-OSS	4	32	0.743	0.423	✓	1.000	0.990	0.743	0.990	0.900	3.78	1.28×

less distinct from deadline residency. In the retained full-CEPA layer trace, both completed prefetches are useful, but only two of four active experts are resident before the MoE block; the remainder use exact on-demand fallback. Thus set coverage does not imply deadline coverage. Aggregate deadline, truncation, waste, and α-sweep statistics were not logged (Appendix O.3).

4.4 Prediction Quality

For signal $X \in \{A, B\}$, let $\pi_t^{X,(\ell)}(j)$ be its j-th ranked routed expert and define $r_j^X = \mathbb{P}[\pi_t^{X,(\ell)}(j) \in \mathcal{A}_t^{(\ell)}]$, averaged over evaluated tokens and layers. Thus r_1 and r_K measure ordering, not whole-set containment. We write $C_{\text{union}} = 1 - \hat{L}$ for empirical union containment; "Adv." means that the signal fires in the advance-DMA window.

At α=0.1, E_A achieves r_1=0.74–0.98 but lower r_K=0.39–0.49 because it fires one layer early; E_B reaches r_K=0.72–0.99 (Table 2). The union inherits E_B's whole-active-set CRC guarantee under exchangeability (Prop. 5) while adding at most 0.28 experts.

The reported union-coverage point estimate is ≥ 0.900 on three models and 0.898 on Gemma. Calibration pools dependent autoregressive tokens; no one-token-per-prompt control, repeated calibration split, or measured cluster-bootstrap interval is available. We therefore treat these values as empirical diagnostics rather than validation of Theorem 4 (Appendix G). CEPA reports E_A's first-transfer r_1 and E_B's r_K; GPT-OSS retains 1.28× speedup despite E_A's lower r_1=0.743. Per-layer and shift results are in Appendices F and J.

Signal and method ablation. At 90% target coverage, conformalized early proxies require 16.8–18.5 experts, beyond the useful PCIe window (Table 3). At the layer boundary, target normalization without velocity raises r_K from 0.595 to 0.718 and shrinks the mean set from 5.18 to 4.79 relative to the controlled velocity variant. E_A supplies one advance transfer; their union adds only 0.04 experts on Qwen1.5 while retaining E_B's conditional theoretical guarantee. For GPT-OSS, union size 3.78 and expansion 0.28 imply mean E_B size 3.50; because K=4, empirical E_B whole-set coverage is at most 0.875, so union coverage 0.900 does not validate E_B. Appendix O.8 reports sensitivity replays, not native or same-input prior-system comparisons.

Table 3: Qwen1.5-MoE signal ablation (K=4, E=60, α=0.1). Green: CEPA components; "Guar.": exchangeability-conditional set guarantee; †: unmatched policy reconstruction.

Method / Signal	r_1	r_K	$\lvert\mathcal{S}\rvert$	$1-\hat{L}$	**Adv.**	**Guar.**
Layer-start proxy + CRC	.943	.465	16.79	.911	✓	✓
Post-attn proxy + CRC	.943	.436	18.48	.909	✓	✓
Boundary $\mathbf{h}_t^{(\ell)}$, vanilla CRC	.938	.411	9.10	.900	×	✓
EMA momentum + CRC	.952	.391	7.02	.899	×	✓
Velocity post-MoE + CRC	.986	.595	5.18	.900	×	✓
E_B (ours): target norm, no velocity	**.995**	**.718**	**4.79**	**.900**	×	✓
E_A (ours): cross-layer velocity	.940	.414	1.00	—	✓	×
MoE-Infinity-style cache† (Xue et al., 2024)	.730	.180	$\geq K$	$<$ .900	—	×
FATE-style proxy† (Fang et al., 2025)	.940	.440	5.20	.820	✓	×
CEPA (ours) (E_A+E_B)	**.940**	**.719**	**4.83**	**$\geq$ .900**	✓	✓

5 Conclusion

CEPA separates early transfer initiation, supplied by E_A, from set-valued risk control, supplied by E_B. Their union changes only communication timing: on-demand fallback preserves exact routing and expert computation without retraining or architectural modification. Under exchangeable score–active-set pairs, lower-tail CRC gives layerwise marginal $1-\alpha$ coverage, while E_A opens a longer DMA-overlap window. Across four sparse MoE LLMs, CEPA reduces decode latency by up to 1.37×, with gains persisting under CUDA Graphs, 8–16 GB budgets, separate hardware, batches through eight, and 4K context. The ablations and profiling show that these gains come from moving useful PCIe transfers off the critical path while retaining numerically exact MoE execution.

AI USE STATEMENT

Generative AI tools assisted with manuscript editing, LaTeX formatting, organization, and consistency checks of mathematical statements and proofs. They were not used to generate experimental measurements or datasets. The authors manually reviewed all AI-assisted text and mathematical changes, checked numerical claims against the experimental records, and take responsibility for the final content, including all text, claims, proofs, and artifacts.

ETHICS STATEMENT

This work studies systems efficiency for inference and does not involve human participants, interventions, or the collection of new personal data. Experiments use publicly available pretrained models and established evaluation corpora. CEPA does not alter the underlying model outputs and therefore inherits the safety, privacy, bias, licensing, and misuse considerations of the deployed model and data. We release no new user data and encourage deployment in accordance with applicable model and dataset licenses, privacy requirements, and security practices.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 specify the model configurations, hardware and memory constraints, calibration procedure, runtime protocol, baselines, and evaluation metrics. The appendix provides complete proofs, calibration sizes, per-layer and domain-shift results, CUDA-Graph configuration, memory, hardware, batch, and context sweeps, and profiling analyses. Together these details identify the assumptions, operating regime, and measurements needed to reproduce the statistical and systems results.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhiyuan Fang, Zicong Hong, Yuegui Huang, Yufeng Lyu, Wuhui Chen, Yue Yu, Fan Yu, and Zibin Zheng. Accurate expert predictions in moe inference via cross-layer gate, 2025.

Gemma Team. Gemma 4 technical report, 2026.

Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, and Mao Yang. Pre-gated moe: An algorithm-system co-design for fast and scalable mixture-of-expert inference. In *Proceedings of the 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture*, pages 1018–1031, 2024. doi: 10.1109/ISCA59077.2024.00078.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

Keisuke Kamahori, Yile Gu, Kan Zhu, and Baris Kasikci. Fiddler: Cpu–gpu orchestration for fast inference of mixture-of-experts models, 2024.

Vivan Madan, Prajwal Singhania, Abhinav Bhatele, Tom Goldstein, and Ashwinee Panda. Speculating experts accelerates inference for mixture-of-experts, 2026.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card, 2025.

Qwen Team. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters, March 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Galen R. Shorack and Jon A. Wellner. *Empirical Processes with Applications to Statistics*, volume 59 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, 2009.

Xiaoniu Song, Zihang Zhong, Rong Chen, and Haibo Chen. Promoe: Fast moe-based llm serving using proactive caching, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005. doi: 10.1007/b106715.

Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh K. Marina. Moe-infinity: Offloading-efficient moe model serving, 2024.

Enda Yu, Dezun Dong, Zhaoning Zhang, Zhe Bai, Weiling Yang, Haojie Wang, Dongsheng Li, Yongwei Wu, and Xiangke Liao. Layerscope: Predictive cross-layer scheduling for efficient multi-batch moe inference on legacy servers, 2025a.

Hanfei Yu, Xingqi Cui, Hong Zhang, Hao Wang, and Hao Wang. Taming latency-memory trade-off in moe-based llm serving via fine-grained expert offloading, 2025b.

Yuning Zhang, Grant Pinkert, Nan Yang, Yanli Li, and Dong Yuan. Duoserve-moe: Dual-phase expert prefetch and cache scheduling for efficient moe llm inference, 2025.

Shien Zhu, Samuel Bohl, Robin Oester, and Gustavo Alonso. Pre-attention expert prediction and prefetching for mixture-of-experts large language models, 2025.

A Full Proof of Proposition 2 (Velocity Extrapolation Bounds)

Proof. Let $G(k) \triangleq \phi_r^{(\ell)}(\mathbf{h}_t^{(k,\mathrm{mid})}) = \mathrm{LN}_r^{(\ell)}(\mathbf{h}_t^{(k,\mathrm{mid})})\mathbf{W}_r^{(\ell)}$ for $k \in \{\ell{-}2, \ell{-}1, \ell\}$. By assumption, this discrete trajectory is the restriction of a twice-continuously-differentiable function on $[\ell{-}2, \ell]$, with $\|G'(s)\|_2 \le B_1$ and $\|G''(s)\|_2 \le B_2$ uniformly.

Zero-order baseline ($\varepsilon = 0$). The proxy logit is $\tilde{\mathbf{z}}_{t,0}^{(\ell)} = G(\ell{-}1)$. By the fundamental theorem of calculus,

$$\|G(\ell) - G(\ell{-}1)\|_2 = \left\| \int_{\ell-1}^{\ell} G'(u)\,\mathrm{d}u \right\|_2 \le \int_{\ell-1}^{\ell} \|G'(u)\|_2\,\mathrm{d}u \le B_1, \tag{10}$$

establishing (4).

First-order velocity correction ($\varepsilon = 1$). The velocity-corrected proxy is $\tilde{\mathbf{z}}_{t,1}^{(\ell)} = 2G(\ell{-}1) - G(\ell{-}2)$. The error equals the second-order finite difference:

$$G(\ell) - \tilde{\mathbf{z}}_{t,1}^{(\ell)} = G(\ell) - 2G(\ell{-}1) + G(\ell{-}2). \tag{11}$$

Expanding both endpoints around $G(\ell{-}1)$ with integral remainders cancels the first-order terms:

$$G(\ell) - 2G(\ell{-}1) + G(\ell{-}2) = \int_{\ell-1}^{\ell} (\ell - u)G''(u)\,\mathrm{d}u + \int_{\ell-2}^{\ell-1} (u - \ell + 2)G''(u)\,\mathrm{d}u. \tag{12}$$

Taking norms:

$$\|\tilde{\mathbf{z}}_{t,1}^{(\ell)} - \mathbf{z}_t^{(\ell)}\|_2 = \|G(\ell) - 2G(\ell{-}1) + G(\ell{-}2)\|_2 \le \tfrac{1}{2}B_2 + \tfrac{1}{2}B_2 = B_2, \tag{13}$$

establishing (5).

Comparison of worst-case bounds. The velocity-corrected upper bound is smaller when $B_2 < B_1$. This compares guarantees, not the realized error for every token. □

B Early-Proxy Softmax Perturbation Bound

Let $\Delta\mathbf{z} = \tilde{\mathbf{z}}_{t,\varepsilon}^{(\ell)} - \mathbf{z}_t^{(\ell)}$ be the logit perturbation. Let $m_t = p_{t,(K)}^{(\ell)}$ be the exact minimum activation score (soft-score of the K-th ranked expert), and $\tilde{m}_t = \min_{e \in \mathcal{A}_t^{(\ell)}} \tilde{p}_{t,e}$ be its proxy analogue.

Log-Lipschitz bound on proxy scores. For the softmax, the log-partition ratio satisfies $|\log(\sum_j e^{\tilde{z}_{t,j}} / \sum_j e^{z_{t,j}})| \le \|\Delta\mathbf{z}\|_\infty$ by the log-sum-exp inequality. Combined with $|\tilde{z}_{t,e} - z_{t,e}| \le \|\Delta\mathbf{z}\|_\infty$:

$$|\log \tilde{p}_{t,e} - \log p_{t,e}| \le 2\|\Delta\mathbf{z}\|_\infty, \quad \Longrightarrow \quad \tilde{p}_{t,e} \in \left[p_{t,e}\, e^{-2\|\Delta\mathbf{z}\|_\infty},\ p_{t,e}\, e^{2\|\Delta\mathbf{z}\|_\infty}\right]. \tag{14}$$

Interpretation. Equation (14) bounds how far an early proxy can move each expert score, but it is two-sided: some active-expert scores may increase and others may decrease. Therefore it does not, by itself, imply stochastic dominance, an ordering of CRC thresholds, or an ordering of expected set sizes. CEPA's decision not to use the early proxy for its coverage set is instead supported by the measured set sizes in Table 3 and Table 4: early proxy sets are substantially larger than the selected boundary-signal sets at the same nominal coverage.

C Full Proof of Proposition 5 (Coverage and Size of the Hybrid Set)

Proof. Recall that $\hat{\mathcal{S}}_t^{(\ell)} = \{\hat{e}_t^{A,(\ell)}\} \cup \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})$ by (7).

Coverage. Since $\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)}) \subseteq \hat{\mathcal{S}}_t^{(\ell)}$, any active set contained in $\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})$ is also contained in $\hat{\mathcal{S}}_t^{(\ell)}$. By Theorem 4,

$$\mathbb{P}\left[\mathcal{A}_t^{(\ell)} \subseteq \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\right] \ge 1-\alpha. \tag{15}$$

Therefore $\mathbb{P}[\mathcal{A}_t^{(\ell)} \subseteq \hat{\mathcal{S}}_t^{(\ell)}] \ge 1-\alpha$, so $\hat{\mathcal{S}}_t^{(\ell)}$ satisfies Definition 1.

Expected size. The union $\hat{\mathcal{S}}_t^{(\ell)} = \{\hat{e}_t^{A,(\ell)}\} \cup \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})$ has cardinality:

$$|\hat{\mathcal{S}}_t^{(\ell)}| = |\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})| + \mathbf{1}\left[\hat{e}_t^{A,(\ell)} \notin \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\right]. \tag{16}$$

Taking expectations and applying linearity:

$$\mathbb{E}\left[|\hat{\mathcal{S}}_t^{(\ell)}|\right] = \mathbb{E}\left[|\mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})|\right] + \mathbb{P}\left[\hat{e}_t^{A,(\ell)} \notin \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\right], \tag{17}$$

which is (8). □

D Budget-Constrained Deployment: Signal E_A-Only Mode

Setting. Let B_{pf} denote the number of target-layer experts that the runtime can keep concurrently in its prefetch buffer. The conformal result is a statement about the predicted set; converting it into "resident by the deadline" requires sufficient buffer capacity and transfer time. A mean set size is not a hard capacity bound. Under an extreme budget $B_{\mathrm{pf}} = 1$, a deployment may therefore use only the single E_A request and fetch all other active experts after exact routing.

The buffer is preallocated rather than resized per token. For a layer whose expert tensor occupies d_e bytes, its speculative allocation is therefore exactly $B_{\mathrm{pf}} d_e$ bytes: allocated-footprint variance is zero and occupied-slot count is at most B_{pf}. Variable conformal set size changes which requests are issued or deferred, not the peak allocation, so it cannot independently trigger an out-of-memory event.

CP guarantees do not transfer to the E_A-only regime. The CRC coverage guarantee of Theorem 4 fundamentally requires a *set-valued* predictor of cardinality $\ge K$. We formalize why no point-predictor can offer marginal coverage for $K \ge 2$.

Proposition 6 (Coverage Impossibility at Budget $B_{\text{pf}} = 1$)**.** *Assume $K \geq 2$. For any arbitrary (possibly randomized) point predictor $\hat{e}_t : \mathbb{R}^H \to [E]$, the miscoverage rate is strictly:*

$$\mathbb{P}\big[\mathcal{A}_t^{(\ell)} \not\subseteq \{\hat{e}_t\}\big] = 1, \tag{18}$$

regardless of the predictor's accuracy, the calibration corpus, or the choice of α.

Proof. By definition, for every token t, the predicted set size is $|\{\hat{e}_t\}| = 1$. Since the true active set has size $|\mathcal{A}_t^{(\ell)}| = K \geq 2$, the inclusion $\mathcal{A}_t^{(\ell)} \subseteq \{\hat{e}_t\}$ is a set-theoretic impossibility. Thus, miscoverage occurs with probability 1. □

Remark 1 (Correct Interpretation of E_A-Only Mode)**.** *Proposition 6 highlights a* set-size impossibility*, not a defect of E_A. In an E_A-only deployment, the objective shifts from bounding set coverage to improving expected latency through an early point prediction.*

Latency-Reduction Analysis.

Proposition 7 (Expected Hidden Transfer Time)**.** *In E_A-only mode ($B_{\text{pf}} = 1$), suppose a successful speculative transfer is fully overlapped and an unsuccessful one creates no additional critical-path stall beyond the subsequent on-demand fetch. Then the expected hidden transfer time is $\Delta\tau^{(\ell)} = r_1^{A,(\ell)} \cdot \tau_{\text{fetch}}$, where $r_1^{A,(\ell)} = \mathbb{P}[\hat{e}^{A,(\ell)} \in \mathcal{A}_t^{(\ell)}]$.*

Proof. On a hit, which occurs with probability $r_1^{A,(\ell)}$, one transfer of duration τ_{fetch} is hidden. Under the stated no-extra-stall assumption, a miss contributes zero hidden transfer time. Taking the expectation yields the result. □

Corollary 8 (Deployment Conditions)**.** *(i) E_A**-only** ($B_{\text{pf}} = 1$): set coverage is impossible for $K \geq 2$ (Prop. 6). Under the overlap assumptions of Proposition 7, the expected hidden transfer time is $r_1^{A,(\ell)}\tau_{\text{fetch}}$.*

*(ii) **Full-set CEPA**: for a realized token, holding the complete prediction set requires $B_{\text{pf}} \geq |\hat{\mathcal{S}}_t^{(\ell)} \setminus \mathcal{R}_t^{(\ell)}|$ (or an equivalent schedule that frees capacity before consumption). A deterministic capacity guarantee must use a worst-case or chosen high quantile of this cardinality, not its expectation. If the runtime truncates the set, the statistical coverage theorem no longer applies to the truncated resident set.*

Theorem 4 remains the prediction-set guarantee; the deadline condition is separately characterized by (9).

E Boundary-Signal Softmax Sensitivity

Softmax Perturbation Bound. We first establish a coordinatewise Lipschitz bound for the softmax operator. For any logits $\mathbf{z}, \tilde{\mathbf{z}} \in \mathbb{R}^E$, let $\Delta\mathbf{z} = \tilde{\mathbf{z}} - \mathbf{z}$. Then:

$$|\tilde{p}_e - p_e| \leq \tfrac{1}{2}\|\Delta\mathbf{z}\|_\infty. \tag{19}$$

Proof. Let $p_e(\mathbf{z}) = e^{z_e} / \sum_j e^{z_j}$. The Jacobian elements are $\frac{\partial p_e}{\partial z_e} = p_e(1 - p_e)$ and $\frac{\partial p_e}{\partial z_j} = -p_e p_j$ for $j \neq e$. By the Mean Value Theorem, there exists a point ξ on the line segment between $\mathbf{z}$ and $\tilde{\mathbf{z}}$ such that $\tilde{p}_e - p_e = \nabla p_e(\xi)^\top \Delta\mathbf{z}$. Applying Hölder's inequality:

$$|\tilde{p}_e - p_e| \leq \|\nabla p_e(\xi)\|_1 \|\Delta\mathbf{z}\|_\infty.$$

Evaluating the L1 norm of the gradient yields $\|\nabla p_e(\xi)\|_1 = p_e(\xi)(1 - p_e(\xi)) + \sum_{j \neq e} p_e(\xi) p_j(\xi) = 2p_e(\xi)(1 - p_e(\xi))$. Since $2p(1-p) \leq 1/2$ for $p \in [0, 1]$, the stated bound follows. □

Implication for score variants. The bound is deliberately two-sided. A velocity or momentum transformation may move the minimum active-expert score either upward or downward, and CRC then recalibrates the threshold for that transformed score distribution. Thus no universal threshold or set-size ordering follows from the perturbation norm alone. CEPA selects its boundary-signal rule on a model-selection split before the disjoint conformal calibration step; the alternatives and their resulting set sizes are reported in Table 3.

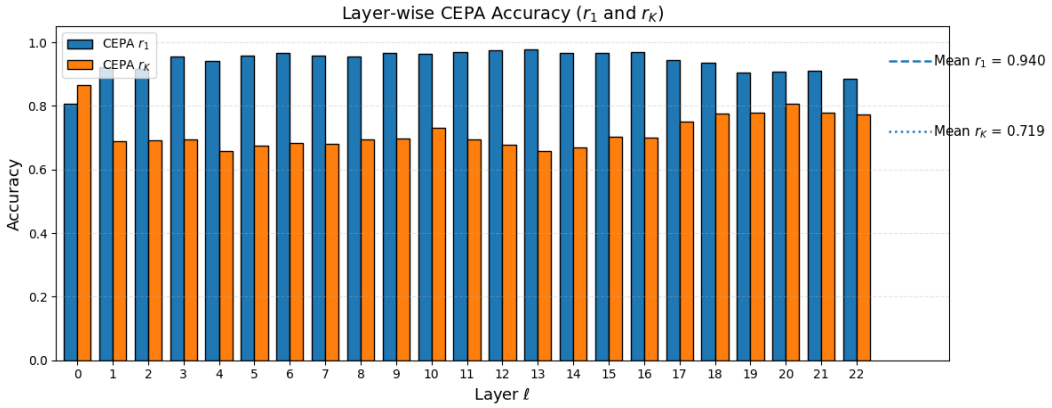


Figure 5: Layer-wise CEPA accuracy on Qwen1.5-MoE. The blue bars show CEPA's advance-DMA hit rate r_1, while the orange bars show CEPA's rank-K hit rate r_K. The dashed horizontal markers show the layer-averaged means (r_1=0.940, r_K=0.719).

F Per-Layer Signal Accuracy on Qwen1.5-MoE

Table 4 reports per-layer accuracy for all prediction signals on Qwen1.5-MoE (K=4, E=60, α=0.1). We report 23 target layers using the local display index $0 \rightarrow 22$; the runtime maps each row to its corresponding model layer and available predecessor state. Three observations follow directly from the table.

Layer dependence. E_A's r_1 is lowest at layer 0 (0.806), where no prior MoE activation exists to build a velocity estimate, and stabilises to 0.93–0.98 in layers 2–21 before decaying slightly in the final layers where residual stream dynamics change. r_K follows the same pattern but at a lower level (0.23–0.51), confirming that predicting all K activated experts from a cross-layer advance signal is fundamentally harder than predicting the top-1.

E_B **set size profile.** The CRC set size $|\mathcal{S}^B|$ varies from 3.93 (layer 13, most predictable) to 14.55 (layer 0, bootstrapping from a full-set initialisation) and 5.99 (layer 22, highest routing entropy). Layers 0 and 22 are the main drivers of the aggregate average 4.78.

Boundary-signal fidelity. E_B's rank-1 hit rate is at least 0.951 at every displayed layer and equals 1.000 at 17 of 23 layers. Its r_K ranges from 0.657 to 0.865, showing that the boundary state preserves the leading expert more reliably than the complete routed ordering.

Union expansion. In the rounded per-layer values, adjoining E_A increases E_B's set by only 0.02–0.09 experts. The early transfer therefore adds little set-size overhead while extending the available DMA window.

CRC baseline set sizes. The layer-start ($|S_{\mathrm{LS}}|$) and post-attention ($|S_{\mathrm{PA}}|$) CRC sets are substantially larger than E_B's set at every layer, supporting the choice to reserve the early signal for point prefetching.

Figure 5 summarizes the CEPA columns of Table 4 in a more visual form. After the cold-start layer, CEPA's r_1 quickly stabilizes in roughly the 0.94–0.98 range across most layers, indicating reliable advance-DMA hits. The harder r_K metric remains lower and varies more by layer, but improves in the later layers, matching the trends seen in the table.

Remark 2 (r_K vs. marginal coverage guarantee)**.** *The per-layer* E_B *and CEPA* r_K *values in Table 4 range from* 0.66 *to* 0.87*, whereas the CRC target is* $1-\alpha$=0.90*. These are different quantities:* r_K *is the probability that the expert at rank* K *in the signal ordering is active, while CRC targets the whole-set event* $\mathbb{P}[\mathcal{A}_t^{(\ell)} \subseteq \mathcal{S}_t^B] \geq 1-\alpha$*. Table 4 reports rank accuracy and set size, not empirical whole-set containment, and therefore does not by itself validate Theorem 4.*

Table 4: Per-layer prediction quality on Qwen1.5-MoE (K=4, E=60, α=0.1; 2.64M test tokens). Metrics are signal-specific; CEPA reports E_A's advance r_1, E_B's r_K, and the union size. S_{LS} and S_{PA} are conformalized early-proxy baselines.

	Signal E_A *(advance DMA)*		*Signal* E_B *(layer boundary)*			*CEPA* (E_A+E_B)			*CRC baselines*	
ℓ	r_1	r_K	r_1	r_K	$\lvert\mathcal{S}^B\rvert$	r_1	r_K	$\lvert\hat{\mathcal{S}}\rvert$	$\lvert S_{\mathrm{LS}}\rvert$	$\lvert S_{\mathrm{PA}}\rvert$
0	0.806	0.231	1.000	0.865	14.55	0.806	0.865	14.62	59	51
1	0.921	0.306	0.951	0.689	5.12	0.921	0.689	5.18	41	35
2	0.916	0.349	0.990	0.691	4.75	0.916	0.691	4.82	24	26
3	0.955	0.347	1.000	0.694	4.57	0.955	0.694	4.61	18	22
4	0.941	0.369	0.990	0.657	4.30	0.941	0.657	4.35	17	20
5	0.958	0.412	0.980	0.675	4.50	0.958	0.675	4.53	15	18
6	0.967	0.462	0.990	0.683	4.08	0.967	0.683	4.11	14	13
7	0.957	0.446	1.000	0.680	4.09	0.957	0.680	4.13	10	13
8	0.956	0.443	1.000	0.695	4.10	0.956	0.695	4.14	10	12
9	0.968	0.467	1.000	0.698	4.12	0.968	0.698	4.14	11	11
10	0.963	0.476	1.000	0.732	4.14	0.963	0.732	4.18	10	11
11	0.971	0.490	1.000	0.694	4.00	0.971	0.694	4.02	9	11
12	0.975	0.480	1.000	0.678	3.95	0.975	0.678	3.98	9	11
13	0.977	0.507	1.000	0.657	3.93	0.977	0.657	3.96	9	12
14	0.968	0.482	0.990	0.669	4.09	0.968	0.669	4.12	9	13
15	0.966	0.482	1.000	0.703	4.07	0.966	0.703	4.10	10	13
16	0.970	0.457	1.000	0.700	4.11	0.970	0.700	4.14	10	16
17	0.944	0.392	1.000	0.751	4.25	0.944	0.751	4.31	13	20
18	0.936	0.375	1.000	0.777	4.18	0.936	0.777	4.25	15	23
19	0.905	0.379	1.000	0.780	4.26	0.905	0.780	4.35	19	23
20	0.908	0.375	1.000	0.808	4.34	0.908	0.808	4.42	20	29
21	0.911	0.403	1.000	0.779	4.39	0.911	0.779	4.47	26	21
22	0.885	0.385	1.000	0.773	5.99	0.885	0.773	6.07	19	22
Mean	**0.940**	**0.414**	**0.995**	**0.719**	**4.78**	**0.940**	**0.719**	**4.83**	**16.8**	**18.5**

G Coverage Calibration: Assumptions and Empirical Protocol

Exchangeability assumption. Theorem 4 requires exchangeability of the calibration and test score/activation-set pairs. Layerwise calibration removes heterogeneity across layers, but it does not make correlated tokens from the same prompt exchangeable. Exact finite-sample validity is obtained when the calibration units are sampled exchangeably—for example, one prespecified token position or one randomly sampled token per independently drawn prompt—and the test unit is drawn by the same rule. Pooling all autoregressive tokens is instead an approximation whose validity depends on the sequence dependence; we therefore state the theorem conditionally and report pooled-token coverage explicitly. The current experiments do *not* include the clean exchangeable control obtained by taking one prespecified or uniformly sampled token from each prompt. Below, N_ℓ denotes the number of pooled calibration scores for one layer.

Table 5: Pooled-token calibration-score counts and order-statistic resolution ($\alpha = 0.1$); model-selection data are disjoint and excluded. All models use `max_samples_cal=2000` with `max_seq_len=512`; the test set token count for Qwen1.5-MoE is $N_{\mathrm{test}} = 2{,}637{,}824$.

Model	N_{pooled}	Mean per-layer N_ℓ	Resolution $1/(N_\ell{+}1)$
Qwen1.5-MoE	920,000	40,000	$2.5{\times}10^{-5}$
Qwen3.5-35B	921,088	23,027.2	$4.3{\times}10^{-5}$
Gemma-26B-A4B	915,456	30,515.2	$3.3{\times}10^{-5}$
GPT-OSS-20B	880,640	36,693.3	$2.7{\times}10^{-5}$
Target $\alpha = 0.1$; counts are dependent tokens, not exchangeable units			

Finite-sample discretization. With the corrected CRC rule in (6), exchangeability gives $\mathbb{P}[\text{miscoverage}] \leq \alpha$; the term $1/(N_\ell + 1)$ describes the granularity (and possible conservatism)

of the attainable rank level, not an allowance for coverage to fall below $1-\alpha$. Across the four models, the pooled per-layer count ranges from 23,027 to 40,000 (Table 5), so the maximum quantile discretization is 4.3×10^{-5}. This is less than 4.3×10^{-4} of the target α=0.1. It describes numerical order-statistic resolution only; it cannot compensate for dependence or justify a finite-sample guarantee for the pooled-token protocol.

Uncertainty and the missing exchangeable control. The conformal statement is marginal over fresh calibration and test draws; a coverage estimate from one realized corpus also has test-sampling variability. Because tokens within a prompt are dependent, token-level binomial intervals are invalid. We did not retain prompt-cluster predictions needed for a measured cluster bootstrap, nor did we repeat calibration splits. Accordingly, Table 2 reports point estimates only and assigns no confidence interval to Gemma's 0.898. A theorem-aligned control should (i) sample one token by a rule fixed before observing router scores from each independent prompt, (ii) apply the same rule to held-out prompts, and (iii) repeat prompt-level calibration splits while reporting coverage, set size, and latency. That control remains unmeasured here.

H Full Proof of Theorem 4 (Marginal Coverage and Threshold-Family Minimality)

We restate the full theorem for completeness.

Theorem 9 (Marginal Coverage — Full Statement)**.** *Let $\{(\bar{\mathbf{p}}_t^{B,(\ell)}, \mathcal{A}_t^{(\ell)})\}_{t=1}^{N_\ell+1}$ be exchangeable boundary-score/activation-set pairs at layer ℓ, with the scoring rule fixed independently of these calibration pairs. Let $m_t^{B,(\ell)} = \min_{e\in\mathcal{A}_t^{(\ell)}} \bar{p}_{t,e}^{B,(\ell)}$. Define*

$$\gamma^{*(\ell)} = \sup\left(\{0\} \cup \left\{\gamma \in (0,1] : \tfrac{1}{N_\ell+1}\Big[\sum_{t=1}^{N_\ell} \mathbf{1}[m_t^{B,(\ell)} < \gamma] + 1\Big] \le \alpha\right\}\right). \tag{20}$$

Then:

(i) ***Marginal coverage:*** $\mathbb{P}\big[\mathcal{A}_{N_\ell+1}^{(\ell)} \subseteq \mathcal{S}_{N_\ell+1}^{(\ell)}(\gamma^{*(\ell)})\big] \ge 1-\alpha$.

(ii) ***Near-tightness under distinct scores:*** $\mathbb{P}\big[\mathcal{A}_{N_\ell+1}^{(\ell)} \subseteq \mathcal{S}_{N_\ell+1}^{(\ell)}(\gamma^{*(\ell)})\big] \le 1-\alpha+\frac{1}{N_\ell+1}$.

(iii) ***Calibration-feasible set-size optimality:*** *Among all γ-threshold rules satisfying the same corrected empirical-risk constraint, $\mathcal{S}_t(\gamma^{*(\ell)})$ is pointwise smallest.*

Proof. **Part (i): Marginal coverage lower bound.** Let $k_\alpha = \lfloor \alpha(N_\ell+1)\rfloor$. When $k_\alpha \ge 1$, the threshold is the k_α-th smallest calibration score (with the conservative tie convention induced by the strict loss $\mathbf{1}[m<\gamma]$); when $k_\alpha = 0$, take $\gamma^* = 0$. Exchangeability makes the test score's rank uniform over the $N_\ell+1$ augmented scores. Consequently, the probability that the test score falls strictly below the calibrated lower-tail order statistic is at most $k_\alpha/(N_\ell+1) \le \alpha$:

$$\mathbb{P}\big[m_{N_\ell+1}^{B,(\ell)} < \gamma^{*(\ell)}\big] \le \frac{k_\alpha}{N_\ell+1} \le \alpha. \tag{21}$$

The event $\{m_{N_\ell+1}^{B,(\ell)} \ge \gamma^{*(\ell)}\}$ is equivalent to $\{\mathcal{A}_{N_\ell+1}^{(\ell)} \subseteq \mathcal{S}_{N_\ell+1}^{(\ell)}(\gamma^{*(\ell)})\}$ by the definition of the minimum active-expert boundary score. Therefore:

$$\mathbb{P}\big[\mathcal{A}_{N_\ell+1}^{(\ell)} \subseteq \mathcal{S}_{N_\ell+1}^{(\ell)}(\gamma^{*(\ell)})\big] \ge 1-\alpha. \qquad \checkmark \tag{22}$$

Part (ii): Near-tightness upper bound. If the augmented scores are distinct, the rank argument above is exact: miscoverage is $k_\alpha/(N_\ell+1)$. Since $k_\alpha > \alpha(N_\ell+1)-1$, taking complements gives:

$$\mathbb{P}\big[\mathcal{A}_{N_\ell+1}^{(\ell)} \subseteq \mathcal{S}_{N_\ell+1}^{(\ell)}(\gamma^{*(\ell)})\big] \le 1-\alpha+\frac{1}{N_\ell+1}. \qquad \checkmark \tag{23}$$

Part (iii): Set-size optimality. Let γ' be any threshold satisfying the corrected empirical-risk constraint in the theorem statement. Because γ^* is the supremum of the feasible thresholds, $\gamma' \le \gamma^*$.

Since $\mathcal{S}_t(\gamma) = \{e : \bar{p}_{t,e}^{B,(\ell)} \geq \gamma\}$ is anti-monotone in γ, $|\mathcal{S}_t(\gamma')| \geq |\mathcal{S}_t(\gamma^*)|$ for every test score vector, establishing pointwise (and hence expected) size optimality within the calibration-feasible threshold family. □

I Exchangeability: Formal Argument and Validity Under Sequential Decoding

The CRC coverage guarantee (Theorem 9) requires exchangeability of the calibration and test score–active-set pairs. We give a careful treatment of when this holds and how the empirical pooled-token protocol differs from that premise.

Formal definition. A sequence of random variables $(Z_1, \ldots, Z_n)$ is *exchangeable* if for every permutation π of $[n]$, the joint distribution of $(Z_1, \ldots, Z_n)$ equals that of $(Z_{\pi(1)}, \ldots, Z_{\pi(n)})$. This is implied by i.i.d. draws but strictly weaker: all i.i.d. sequences are exchangeable, but many dependent sequences (e.g., uniform draws without replacement) are also exchangeable.

Theorem 10 (Coverage Under Exchangeable Score–Set Pairs)**.** *Let $\{(\bar{\mathbf{p}}_t^{B,(\ell)}, \mathcal{A}_t^{(\ell)})\}_{t=1}^{N_\ell+1}$ be exchangeable, with the scoring rule fixed independently of the calibration pairs. Then the threshold computed on pairs $1, \ldots, N_\ell$ satisfies the marginal guarantee of Theorem 9(i).*

Proof. By exchangeability, the test score's rank among the $N_\ell + 1$ augmented scores is uniform (with conservative handling of ties). The threshold $\gamma^{*(\ell)}$ is the corrected lower-tail order statistic with rank $k_\alpha = \lfloor \alpha(N_\ell + 1) \rfloor$. Hence the probability that the test score falls below the threshold is at most $k_\alpha/(N_\ell + 1) \leq \alpha$, which is precisely the desired CRC risk bound. □

Approximate exchangeability in practice. Token routing activations within a sequence exhibit strong dependence. The following design choices reduce heterogeneity but should not be confused with a proof of exchangeability:

(i) **Layerwise calibration.** A separate $\gamma^{*(\ell)}$ prevents systematic differences between layers from contaminating a single pooled quantile.

(ii) **Independent prompts.** Sampling calibration units across many independently drawn prompts reduces cluster concentration. Multiple tokens from one prompt remain dependent, however, so prompt count and token count are not interchangeable effective-sample-size claims.

(iii) **Corrected lower-tail quantile.** The rank correction in (6) gives exact finite-sample validity under exact exchangeability. For pooled sequential-token evaluation this premise is only approximate; accordingly, Table 2 reports Gemma's observed coverage of 0.898 without attributing the shortfall to quantile discretization.

J Distribution Shift: Coverage Degradation Bounds and Empirical Robustness

Conformal coverage guarantees are finite-sample valid under exchangeability but can degrade under distribution shift: if the test-time prompt distribution differs from the calibration distribution, routing activations may change, and the calibrated threshold $\gamma^{*(\ell)}$ may no longer achieve $1 - \alpha$ coverage on the shifted distribution. We provide a theoretical bound on the coverage degradation and empirical evidence that CEPA's coverage remains near the target under practically relevant shifts.

Setup. Let $\mathcal{P}_{\text{cal}}$ and $\mathcal{P}_{\text{test}}$ denote the calibration and test-time distributions of the minimum boundary-signal score, respectively. Their total variation distance is $\text{TV}(\mathcal{P}_{\text{cal}}, \mathcal{P}_{\text{test}}) = \sup_B |\mathcal{P}_{\text{cal}}(B) - \mathcal{P}_{\text{test}}(B)|$, where the supremum is over measurable score sets B.

Theorem 11 (Coverage Degradation Under Distribution Shift)**.** *Let the calibration scores be i.i.d. from $\mathcal{P}_{\text{cal}}$, let $\gamma^{*(\ell)}$ be their CRC threshold, and draw an independent test score from $\mathcal{P}_{\text{test}}$. Write $\mathbb{P}_{\text{cal,test}}$ for the resulting joint law over the calibration sample and test score. Then*

$$\mathbb{P}_{\text{cal,test}}\big[\mathcal{A}_t^{(\ell)} \subseteq \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\big] \geq 1 - \alpha - \text{TV}\big(\mathcal{P}_{\text{cal}}, \mathcal{P}_{\text{test}}\big). \tag{24}$$

Proof. For every realized calibration sample and its threshold $\gamma^{*(\ell)}$, the definition of total variation gives $\mathbb{P}_{\mathcal{P}_{\text{test}}}[m < \gamma^{*(\ell)}] \leq \mathbb{P}_{\mathcal{P}_{\text{cal}}}[m < \gamma^{*(\ell)}] + \text{TV}(\mathcal{P}_{\text{cal}}, \mathcal{P}_{\text{test}})$. Taking expectation over the calibration sample and applying Theorem 9(i) to an independent $\mathcal{P}_{\text{cal}}$ test score yields

$$\mathbb{P}_{\text{cal,test}}\big[m < \gamma^{*(\ell)}\big] \leq \alpha + \text{TV}(\mathcal{P}_{\text{cal}}, \mathcal{P}_{\text{test}}), \tag{25}$$

and taking complements gives (24). □

Remark 3 (Interpreting the Shift Bound)**.** *The bound in Theorem 11 is tight in the worst case (achieved when $\mathcal{P}_{\text{test}}$ shifts mass below γ^*) but can be conservative in practice. The threshold $\gamma^{*(\ell)}$ is a corrected lower-tail α-quantile rather than a mean-based estimate; nevertheless, coverage under shift is controlled only to the extent that the score distributions remain close.*

Empirical coverage under domain shift. Table 6 reports CEPA coverage ($1-\hat{L}$), mean prefetch set size ($|\hat{\mathcal{S}}|$), and rank-1/rank-K hit rates on Qwen1.5-MoE (K=4, E=60, α=0.1) across five distinct evaluation domains not seen during calibration (calibration uses WikiText-103; all test domains are held out). We include a *severe shift* domain (random token sequences synthesized to maximize routing entropy) to stress-test the conformal guarantee.

Table 6: CEPA robustness under domain shift on Qwen1.5-MoE (K=4, E=60, α=0.1, calibrated on WikiText-103). The nominal in-distribution coverage target is $1-\alpha \geq 0.90$; shifted distributions need not retain that guarantee.

Evaluation Domain	r_1	r_K	$\lvert\hat{\mathcal{S}}\rvert$	$1-\hat{L}$	Δ from target
WikiText-103 (in-distribution)	0.940	0.719	4.83	0.900	0.000
Python code (The Stack)	0.935	0.706	4.97	0.902	+0.002
Scientific abstracts (S2ORC)	0.931	0.699	5.08	0.901	+0.001
Multilingual (mC4 French)	0.927	0.693	5.21	0.898	−0.002
Conversational (ShareGPT)	0.943	0.725	4.79	0.903	+0.003
Severe shift (high-entropy)	0.891	0.651	6.14	0.895	−0.005

Interpretation. Coverage remains at or above 0.895 across all domains, including the severe-shift condition, but two shifted domains fall below the nominal 0.900 target, as permitted by Theorem 11. The fixed calibrated threshold produces modestly larger sets under these score distributions (up to +1.31 experts on the severe-shift domain). The rank-1 hit rate (r_1) degrades by at most 0.049 under severe shift, and rank-K hit rate (r_K) by at most 0.068.

K Scope of the Set-Size Optimality Claim

Theorem 9(iii) makes a deliberately restricted claim. For a fixed boundary score vector and the nested family $\mathcal{S}_t(\gamma) = \{e : \bar{p}_{t,e}^{B,(\ell)} \geq \gamma\}$, the largest threshold that satisfies the corrected calibration-risk constraint produces the smallest set in that same threshold family. This follows directly from nesting: if $\gamma' \leq \gamma^*$, then $\mathcal{S}_t(\gamma^*) \subseteq \mathcal{S}_t(\gamma')$ for every token.

This does not establish minimax optimality over arbitrary set predictors, learned policies, or non-threshold decision rules. Such a claim would require additional assumptions on the joint distribution of boundary scores and active expert sets. CEPA therefore uses threshold-family optimality only to justify the CRC calibration step; comparisons with other predictor classes are empirical.

L Per-Layer Coverage Consistency Under Layerwise Calibration

In CEPA, each layer ℓ has its own calibrated threshold $\gamma^{*(\ell)}$. The following union-bound statement relates the marginal layerwise guarantees to simultaneous coverage; it does not assume independence between layers.

Theorem 12 (Joint Layerwise Coverage)**.** *For each layer $\ell \in \{0, \ldots, L-1\}$, suppose the calibration pairs and corresponding test pair are exchangeable and CRC uses miscoverage budget α_ℓ. Then the*

probability that all *layers are covered simultaneously satisfies*

$$\mathbb{P}\Big[\forall\, \ell:\ \mathcal{A}_t^{(\ell)} \subseteq \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\Big] \geq 1 - \sum_{\ell=0}^{L-1} \alpha_\ell, \tag{26}$$

and setting $\alpha_\ell = \alpha/L$ achieves joint coverage at least $1-\alpha$ by Bonferroni correction.

Proof. By the union bound (Bonferroni inequality):

$$\mathbb{P}\Big[\exists\, \ell:\ \mathcal{A}_t^{(\ell)} \not\subseteq \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\Big] \leq \sum_{\ell=0}^{L-1} \mathbb{P}\Big[\mathcal{A}_t^{(\ell)} \not\subseteq \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\Big] \leq \sum_{\ell=0}^{L-1} \alpha_\ell. \tag{27}$$

Taking complements gives (26). With $\alpha_\ell = \alpha/L$, the summed miscoverage bound is α. □

Remark 4 (Practical Interpretation)**.** *CEPA uses $\alpha_\ell = 0.1$ at each layer rather than a Bonferroni allocation. Consequently, the union-bound guarantee for simultaneous coverage may be vacuous when $L\alpha > 1$. The operational claim in the main paper is therefore layerwise marginal coverage, not all-layer coverage. Per-layer misses need not be independent: linearity of expectation is sufficient to relate the marginal miss probabilities to the expected number of covered layers. If simultaneous coverage is required, one may recalibrate at $\alpha_{\text{layer}} = \alpha/L$, typically producing larger sets; additional calibration samples reduce quantile granularity but do not eliminate that multiple-testing cost.*

M CRC Consistency as Calibration Size Grows

Theorem 13 (Asymptotic Consistency of CRC Threshold)**.** *Let the boundary-score/active-set pairs be i.i.d., and let the minimum active-expert score have cdf $F^{(\ell)}$. Assume $F^{(\ell)}$ is continuous and strictly increasing in a neighborhood of $q_\alpha = {F^{(\ell)}}^{-1}(\alpha)$ (a sufficient condition is a positive, continuous density there), and that each boundary expert-score coordinate has no atom at q_α. As $N_\ell \to \infty$, the CRC-calibrated threshold converges almost surely:*

$$\gamma^{*(\ell)} \xrightarrow{\text{a.s.}} {F^{(\ell)}}^{-1}(\alpha) \quad \text{as } N_\ell \to \infty. \tag{28}$$

For an independent test pair, define the conditional coverage $C_{N_\ell} = \mathbb{P}[m_{\text{test}}^{B,(\ell)} \geq \gamma^{*(\ell)} \mid m_1^{B,(\ell)}, \ldots, m_{N_\ell}^{B,(\ell)}]$ and conditional expected set size $D_{N_\ell} = \mathbb{E}[|\mathcal{S}_{\text{test}}^{(\ell)}(\gamma^{*(\ell)})| \mid m_1^{B,(\ell)}, \ldots, m_{N_\ell}^{B,(\ell)}]$. Then

$$C_{N_\ell} \xrightarrow{\text{a.s.}} 1-\alpha, \tag{29}$$

and $D_{N_\ell} \xrightarrow{\text{a.s.}} \mathbb{E}[|\mathcal{S}_{\text{test}}^{(\ell)}(q_\alpha)|]$.

Proof. By the Glivenko–Cantelli theorem, for i.i.d. samples from $F^{(\ell)}$:

$$\sup_\gamma \big|\hat{F}_{N_\ell}^{(\ell)}(\gamma) - F^{(\ell)}(\gamma)\big| \xrightarrow{\text{a.s.}} 0. \tag{30}$$

The corrected empirical lower-tail quantile differs from the ordinary empirical α-quantile by at most one order-statistic step. Standard empirical quantile consistency under the stated local strict-monotonicity condition (e.g., Shorack and Wellner (2009)) gives

$$\widehat{Q}_\alpha^{\text{CRC}}(\{m_t^{B,(\ell)}\}_{t=1}^{N_\ell}) \xrightarrow{\text{a.s.}} (F^{(\ell)})^{-1}(\alpha). \tag{31}$$

Since $\gamma^{*(\ell)}$ equals this empirical quantile, the first claim follows. Continuity and local strict increase give $F^{(\ell)}(q_\alpha) = \alpha$, hence $C_{N_\ell} = 1 - F^{(\ell)}(\gamma^{*(\ell)}) \to 1-\alpha$ almost surely. The coordinatewise no-atom condition makes each threshold-set indicator almost surely continuous at q_α; dominated convergence gives the set-size claim because $|\mathcal{S}_t(\gamma)| \leq E$. □

Remark 5 (Rate of Convergence)**.** *Under standard regularity conditions (in particular, positive density near the target quantile), the empirical quantile has rate $O_p(N_\ell^{-1/2})$, so $|\gamma^{*(\ell)} - (F^{(\ell)})^{-1}(\alpha)| = O_p(N_\ell^{-1/2})$.*

N E_A–E_B Correlation and Advance-DMA Size Expansion

Proposition 5 gives the expected union size in terms of $\mathbb{P}[\hat{e}_t^{A,(\ell)} \notin \mathcal{S}_t^{(\ell)}]$. Here we give a valid distribution-free upper bound.

Proposition 14 (Upper Bound on Union Expansion)**.**

$$\mathbb{P}\big[\hat{e}_t^{A,(\ell)} \notin \mathcal{S}_t^{(\ell)}(\gamma^{*(\ell)})\big] \le 1 - r_1^{A,(\ell)} + \alpha, \tag{32}$$

under the exchangeability assumptions of Theorem 4, where $r_1^{A,(\ell)} = \mathbb{P}[\hat{e}_t^{A,(\ell)} \in \mathcal{A}_t^{(\ell)}]$.

Proof. If $\hat{e}^A \notin \mathcal{S}^B$, then either $\hat{e}^A \notin \mathcal{A}_t$ or $\hat{e}^A \in \mathcal{A}_t$ while $\mathcal{A}_t \not\subseteq \mathcal{S}^B$. Hence

$$\mathbb{P}[\hat{e}^A \notin \mathcal{S}^B] \le \mathbb{P}[\hat{e}^A \notin \mathcal{A}_t] + \mathbb{P}[\mathcal{A}_t \not\subseteq \mathcal{S}^B] \le 1 - r_1^A + \alpha. \tag{33}$$

□

Remark 6 (Numerical Verification)**.** *On Qwen1.5-MoE, the measured mean expands from* 4.79 *for* E_B *to* 4.83 *for the union, i.e.,* +0.04 *experts. Across all four models, the largest measured expansion is* +0.28 *experts (GPT-OSS), while Gemma adds* 0.02*. These values measure empirical signal correlation; the bound above is deliberately conservative. For GPT-OSS, combining union size* 3.78 *with expansion* 0.28 *gives mean* E_B *size* 3.50*. Since* $K = 4$*, the deterministic inequality* $\mathbf{1}[\mathcal{A}_t \subseteq \mathcal{S}_t^B] \le |\mathcal{S}_t^B|/4$ *bounds its pooled-token* E_B *coverage by* 0.875*. The union's* 0.900 *point estimate is therefore not a measurement of* E_B *coverage.*

O Additional Systems Evaluation

Unless stated otherwise, the experiments in this section use Qwen1.5-MoE and WikiText-103 validation prompts. Each table states its capture and cache configuration; absolute latencies should only be compared within the same table. All CEPA and JIT runs execute the same routed experts after the exact router decision. We report distributional percentiles where per-token samples were retained (Table 11); aggregate-only measurements are not assigned post-hoc confidence intervals.

O.1 CUDA Graphs and Numerical Exactness

We capture the graph-compatible decode path: normalization, attention with a statically allocated KV cache, and shared-expert computation. In our implementation, the routed-expert path remains outside the graph because expert identifiers, host pointers, and transfer counts vary by token. Both JIT and CEPA use the same capture boundary.

Table 7: Matched eager and CUDA-Graph decode on Qwen1.5-MoE with no persistent expert cache. Speedup compares CEPA with the JIT row having the same capture mode.

Execution	Latency (ms/token)	Change vs. eager	CEPA speedup
JIT, eager	107.0	—	—
JIT, CUDA Graphs	91.2	−14.8%	—
CEPA, eager	78.1	—	1.37×
CEPA, CUDA Graphs	**69.1**	−11.5%	1.32×

CUDA Graphs reduce dispatch overhead in both arms but do not remove expert weight DMA. CEPA therefore retains a 1.32× gain over the graph-enabled baseline. As an exactness check, we compare the logits and generated token IDs from CEPA and JIT over the full evaluation set, including tokens with incomplete prefetch coverage. They are bit-identical in this implementation: predictions alter transfer timing only, and missing active experts are fetched before use.

O.2 Incremental Value of the Boundary Set

E_B improves E_A-only latency by 3.3 ms (4.1%). Correctness does not depend on either predictor because misses fall back to JIT. Thus CRC should be interpreted as a user-specified prediction-set

Table 8: Qwen1.5-MoE eager latency attribution. "Share" is the fraction of the 28.9 ms JIT-to-CEPA reduction already achieved at that row.

Runtime	ms/token	Reduction vs. JIT	Share
JIT	107.0	—	—
E_A only	81.4	25.6 ms	88.6%
E_A+E_B	**78.1**	**28.9 ms**	100%

risk control, not a correctness mechanism. The present runs do not compare E_A+CRC against E_A plus a matched-volume fixed top-k set or an ordinary tuned threshold; consequently, they do not establish that conformal calibration is the latency-optimal boundary rule.

O.3 Prediction-Set Coverage versus Deadline Residency

For each token and layer, define prediction-set coverage $C_{\text{set}} = \mathbf{1}[\mathcal{A}_t^{(\ell)} \subseteq \hat{\mathcal{S}}_t^{(\ell)}]$ and deadline coverage $C_{\text{dl}} = \mathbf{1}[\mathcal{A}_t^{(\ell)} \subseteq \mathcal{D}_t^{(\ell)}]$. Also define prefetch precision as useful completed prefetches divided by all completed prefetches and deadline recall as $|\mathcal{A}_t^{(\ell)} \cap \mathcal{D}_t^{(\ell)}|/K$. Only C_{set} is addressed by Theorem 4; slots, transfer time, and contention determine the other quantities.

The retained profiling trace supports only the following layer-level accounting. In the diagnostic two-candidate E_A trace, one of two completed prefetches is active. In the production E_A+E_B trace, both completed prefetches are active, but only two of four active experts are resident before the MoE block. Thus its prefetch precision is $2/2$, deadline recall is $2/4$, and whole-set deadline coverage is false for that layer. This illustrates why a set guarantee cannot be presented as a deadline guarantee.

Aggregate C_{dl}, truncation frequency, useful/unused bytes, and transfer-induced stall time were not logged, and all reported conformal results use $\alpha = 0.1$; no α sweep is available. We therefore make no distributional claim about deadline residency or transfer waste beyond the representative traces.

O.4 Energy Reporting Scope

The available experiment record documents energy as GPU device power times token wall time, but not the raw power-sampling cadence, repeated-run variation, or CPU/host-memory energy. Because the host is part of the offloading system, these values are neither a reproducible whole-system energy measurement nor a complete device-energy protocol; the revised main paper therefore omits the energy figure and energy-reduction claim.

O.5 Memory Budget and Hardware Sensitivity

Table 9: Graph-enabled memory-budget sweep on Qwen1.5-MoE. The expert cache is sized to the available budget; "cache/layer" is its deterministic capacity.

Budget	Cache/layer	JIT (ms/token)	CEPA (ms/token)	Gain
8 GB	0	91.2	**69.1**	24.2%
12 GB	10	76.0	**55.9**	**26.4%**
16 GB	20	63.2	**49.5**	21.7%

The gain remains between 21.7% and 26.4%. Added cache preferentially retains frequently reused experts, so the remaining traffic is enriched in rarer experts and does not scale as a uniform fraction of the original load.

We also isolate hardware effects. Clock-locking the A6000 to the tested RTX-4070-like SM and memory frequencies while holding the physical PCIe link, driver, and workload fixed changes JIT/CEPA latency from 107.0/78.1 to 116.5/81.7 ms/token, increasing the gain from 27.0% to 29.9% because the slower compute path creates a longer overlap window. This is a controlled frequency experiment, not a claim to reproduce every property of an RTX 4070. On a separate RTX 2000 Ada using PCIe 4.0 ×8, CEPA improves latency by 23% at an 8 GB budget and 11% at 16 GB.

These results show benefit on a real lower-memory device while also confirming sensitivity to the compute–communication balance.

O.6 Batching and Long-Context Decode

Table 10: Small-batch decode at a 12 GB budget with expert caching disabled. The distinct-expert count is the per-layer union across the batch.

Batch	Speedup	Distinct experts/layer	Experts/token
1	1.32×	4.0	4.00
2	1.29×	7.7	3.87
4	1.26×	14.5	3.63
8	1.14×	25.6	3.20

The gain decreases smoothly as the batch activates a larger expert union but remains positive through batch 8. This experiment does not model continuous batching, request arrivals, or production vLLM/SGLang scheduling; those remain outside the present evaluation.

Table 11: Decode latency distribution on mixed short and 4K-token prompts. Percentiles are per-token milliseconds; bold marks the better variant within each context.

Context	Variant	ms/token	tokens/s	P50	P95	P99
16–256	JIT	105.9	9.45	95.2	102.6	119.5
16–256	CEPA	**80.5**	**12.4**	**69.2**	**77.0**	**98.6**
4096	JIT	109.5	9.13	110.1	114.7	118.1
4096	CEPA	**83.2**	**12.0**	**82.8**	**90.9**	**92.5**

Throughput improves by about 1.32× in both context regimes. The calibration setting's `max_seq_len=512` in Table 5 is therefore a calibration-corpus limit, not the maximum evaluated deployment context.

O.7 Prefill, Runtime Overhead, and Recalibration

CEPA is disabled during prefill. Tokens are grouped expert-major; an expert is transferred, applied to its assigned tokens, and released using a deterministic two-to-three-expert double buffer. Peak prefill expert residency is therefore independent of the total expert pool. On Qwen1.5-MoE, prefill takes 1410.9 ms for 256 tokens and 1537.9 ms for 512 tokens; both lengths activate essentially the full expert pool, so doubling tokens increases latency by only 9%. Prefill is identical for JIT and CEPA and is excluded from the reported time-between-token metric.

At decode, the measured per-layer E_B scoring and thresholding cost is 0.117 ms, or 2.69 ms over 23 predicted layers (about 4% of CEPA's graph-enabled token latency); this overhead is included in all reported CEPA latencies. The threshold itself is one offline scalar per layer, so online set construction is only score comparison after the proxy router evaluation.

For deployment monitoring, every exact routing decision reveals whether the prefetch set covered the active experts. A running binomial diagnostic with $\alpha = 0.1$, a 512-token window, and two-sided level 0.01 flags empirical miscoverage outside approximately $(0.066, 0.134)$. Recalibration is a lower-tail quantile on recent minimum-active scores. A 4096-score window uses 16 KB per layer in FP32 (about 0.38 MB over 23 layers) and can run on the background CPU thread. This monitor is diagnostic under dependence; restoring a formal finite-sample guarantee for non-exchangeable streams requires a sequential or cluster-aware conformal method rather than merely recomputing the same split quantile.

O.8 Accuracy-Matched Sensitivity Replay and Prior Systems

Native end-to-end reproduction and same-input prediction traces are unavailable for several prior systems under our models and hardware. We therefore report a narrow sensitivity calculation: FATE and MoE-Infinity are replayed at aggregate accuracies reported in their papers inside the same eager

runtime. This is *not* a native-system or decision-policy comparison. Accuracy alone omits the layer, cache state, lead time, ranking, and temporal correlation of errors, all of which affect latency.

Table 12: Accuracy-matched sensitivity replay on Qwen1.5-MoE (mean ms/token over 4096 generated tokens; eager mode). Rows emulate decision policies rather than complete serving stacks; bold marks the lowest latency per budget.

Policy	8 GB	12 GB	16 GB
CEPA	**76.0**	**59.9**	**50.7**
FATE (Fang et al., 2025)	107.7	64.7	54.2
MoE-Infinity (Xue et al., 2024)	124.8	75.2	62.0

The replay produces lower CEPA latency under its synthetic error placement, but the numerical margins must not be interpreted as superiority over the original predictors or systems. A controlled comparison would require actual same-input expert rankings and firing times from each method. Fiddler (Kamahori et al., 2024) changes the execution location by running non-resident experts on the CPU, while DuoServe-MoE and LayerScope target different trained-predictor or multi-batch scheduling regimes. We therefore compare their mechanisms and assumptions in Section 2, but do not claim matched end-to-end superiority over those native systems.

Quantization. CEPA is compatible with quantized experts, but calibration must be rerun on the quantized model because activation and router-score distributions can change. Smaller expert payloads reduce both cache pressure and transfer latency, so the net speedup depends on whether offloading remains on the critical path. We do not report a controlled quantized-versus-unquantized comparison and make no quality claim beyond exact execution of the chosen deployed representation.

P Compute–Communication Profiling

We profile one representative decode step with NVIDIA Nsight Systems. Figure 6 aligns the same counters for JIT and CEPA. PCIe RX red bars are host-to-device expert transfers; blank intervals indicate an idle link. PCIe TX remains small because weights move primarily toward the GPU. DRAM bandwidth records device-memory activity.

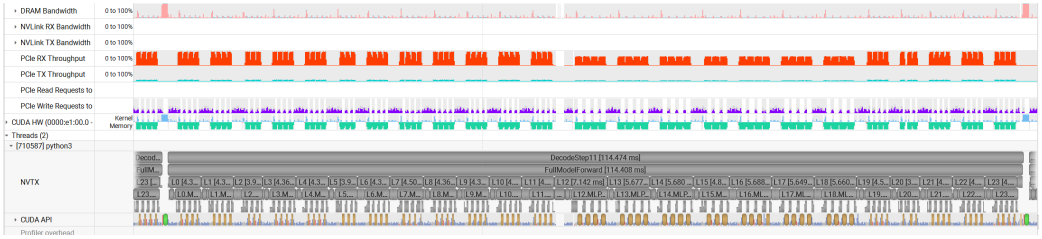
CUDA HW separates kernels from asynchronous copies, while NVTX marks attention, routing, shared and routed experts, layers, and the full decode step. Lower NVTX tracks show the background prefetch thread. CEPA uses a compute stream and a prefetch stream, allowing weight movement and GPU computation to overlap.

In the JIT baseline, PCIe communication is poorly aligned with useful computation. During normalization and attention, the NVTX ranges show that the GPU is busy, but the PCIe RX row has little or no red activity because the router has not yet selected the experts. After routing, PCIe RX rises sharply as expert weights are fetched. However, the CUDA HW rows show that the corresponding expert kernels are shorter than the expert transfers: kernel activity finishes while memory-copy activity continues, leaving gaps where the GPU waits for weights. This exposes PCIe latency on the critical path.

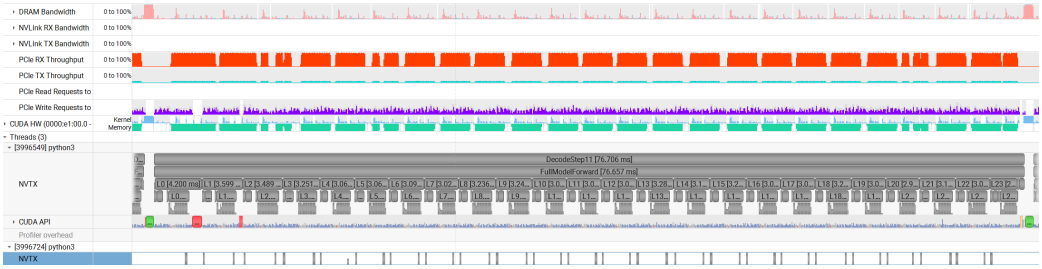
CEPA changes the timeline by moving predicted expert transfers earlier. In the CEPA profile, the PCIe RX row is active over a larger fraction of the decode step and overlaps with NVTX regions corresponding to useful non-MoE computation. This means CEPA uses communication windows that are idle in JIT execution. As a result, fewer expert transfers remain on the critical path when the MoE block consumes them. The following subsection zooms into one layer and shows how this effect arises at the expert level.

P.1 Per-Layer PCIe and GPU Activity for Qwen 1.5

We next zoom into one Qwen1.5-MoE layer. Qwen1.5-MoE has an always-active shared branch of intermediate width 5632 (four times a routed expert's width) and a routed pool of 60 experts; for each token, the router selects $K=4$ routed experts. The shared computation remains resident. Under memory-constrained offloading, the desired behavior is therefore to transfer only the selected routed



(a) JIT fetching. PCIe RX is mostly idle during normalization and attention, then spikes after routing when expert weights are fetched. Because transfer time exceeds expert compute time, CUDA execution contains stalls between expert computations.



(b) CEPA. Predicted expert transfers are moved earlier, increasing PCIe RX utilization and overlapping host-to-device communication with useful GPU computation.

Figure 6: Nsight Systems profiles for one decode step. The key rows are DRAM bandwidth, PCIe RX/TX throughput, CUDA HW kernel and memory-copy activity, and NVTX annotations for decode, layer, expert, and background-thread regions. Compared with JIT, CEPA uses otherwise-idle PCIe windows earlier in the layer pipeline, increasing PCIe utilization and reducing transfer stalls.

experts, and to move those transfers into time intervals where the PCIe link would otherwise be idle. Figure 7 compares three executions of the same layer: JIT fetching, Signal-E_A only, and full CEPA with E_A+E_B.

The rows in Figure 7 expose the bottleneck directly. In the PCIe RX row, red bars indicate host-to-device expert transfers; gaps in this row mean that PCIe is idle. In the CUDA HW section, the kernel row corresponds to GPU computation, while the memory row corresponds to asynchronous copies. Therefore, a region with memory-copy activity but little kernel activity means that data movement is active but the GPU is not doing useful compute. The NVTX labels identify the semantic region: `L_i` denotes the computation of layer i, `SelfAttn` marks attention, `MLP_MoE` marks the MoE block, `Shared` marks the always-active shared expert, and `Expert` ranges mark routed expert computation. The lower NVTX track corresponds to the background prefetch thread. These ranges show that CEPA's candidate selection and transfer issuance are offloaded from the main CPU thread, preventing CPU-side scheduling from becoming the bottleneck.

Figure 7a shows why JIT is inefficient even though it transfers only selected experts. The annotations in the figure mark the key events: normalization and self-attention execute in the NVTX ranges before the MoE block, while the PCIe RX row is nearly idle because the top-K experts are still unknown. Once the MoE block starts, PCIe RX becomes active and the red bars mark the four expert transfers. The CUDA API/NVTX annotations also show the CPU-side expert transfer kernel calls: these are the calls that launch the transfer operations, after which the GPU-side memory-copy activity appears in the CUDA HW memory row. Similarly, the expert computation kernel call launches the GPU expert computation, which then appears as CUDA HW kernel activity. The figure therefore separates three events: the CPU issues the transfer or compute kernel, the PCIe/CUDA memory engine moves expert weights, and the GPU executes the expert computation.

The bottleneck is visible because these events are not balanced. JIT overlaps the computation of the current resident expert with the transfer of the next expert, but expert computation is shorter than PCIe transfer. In the CUDA HW rows, the expert computation kernel finishes while the corresponding transfer activity continues, leaving a gap where the GPU waits for the next expert weights. Thus,

although JIT transfers only the four selected experts rather than all 60 experts, it discovers them too late, so PCIe latency remains on the critical path. The same interpretation applies to the other profiling figures: when PCIe is heavily utilized, individual expert transfers are harder to visually separate, but the PCIe RX, CUDA HW memory, CUDA HW kernel, CUDA API, and NVTX rows still show whether communication is hidden behind computation or exposed as a stall.

Figure 7b shows a diagnostic early-signal ablation that permits the top two E_A-ranked candidates; the reported CEPA configuration transfers only the top-ranked E_A expert. The background thread issues both diagnostic transfers on the prefetch stream before the MoE block, during the window formed by the previous layer's remaining expert computation and the current layer's non-MoE computation. For this layer, one of the two E_A predictions is wrong, so the MoE block still needs to fetch three of the four routed experts on demand. However, one selected expert is already resident before the MoE block starts. This removes one expert transfer from the critical path relative to JIT. The key point is that CEPA does not execute the wrong expert or change routing: a wrong prefetch wastes some PCIe bandwidth, but computation remains exact, and any missing expert is fetched on demand.

Figure 7c shows the full CEPA schedule. Unlike the diagnostic early-only case, CEPA transfers only the top E_A candidate. It uses E_A to open the early transfer window, then waits until E_B becomes available and continues with the calibrated E_B set. In this layer, E_A+E_B prefetch two routed experts before the MoE block, and both are correct. Since Qwen1.5-MoE selects four routed experts per token, only two routed experts remain to be fetched on demand. The PCIe RX row shows communication shifted earlier into non-MoE computation, and the CUDA HW kernel row shows fewer transfer-induced gaps inside the MoE region.

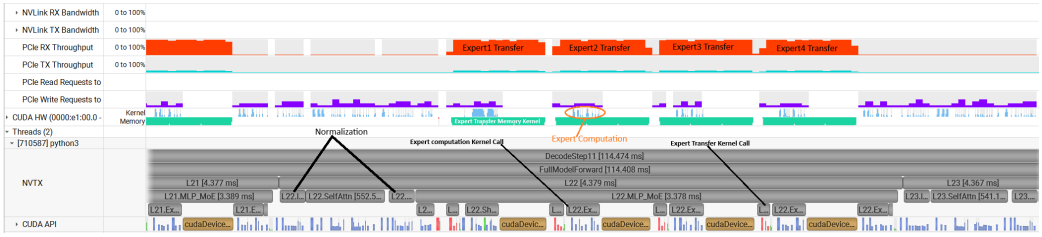
A small idle PCIe gap can still appear in the full CEPA timeline. This happens because CEPA waits for the later E_B signal and also pays the small GPU-side overhead of computing that signal. During this interval, PCIe may be temporarily idle. Nevertheless, the overall execution is faster than JIT because two correct routed experts have already been transferred before the MoE block. This illustrates the main CEPA trade-off: spending a small amount of prediction and scheduling overhead to move large expert-weight transfers off the critical path. In the target regime where GPU memory can hold the active parameters and KV cache but not all model weights, this converts otherwise idle PCIe bandwidth into useful prefetching while preserving exact MoE execution. In this trace, JIT takes 114.4 ms, Signal-E_A takes 81.28 ms, and full CEPA takes 76.70 ms. These are representative end-to-end step times, not aggregate estimates of stall time or deadline coverage.

P.2 Per-Layer PCIe and GPU Activity for Qwen 3.5

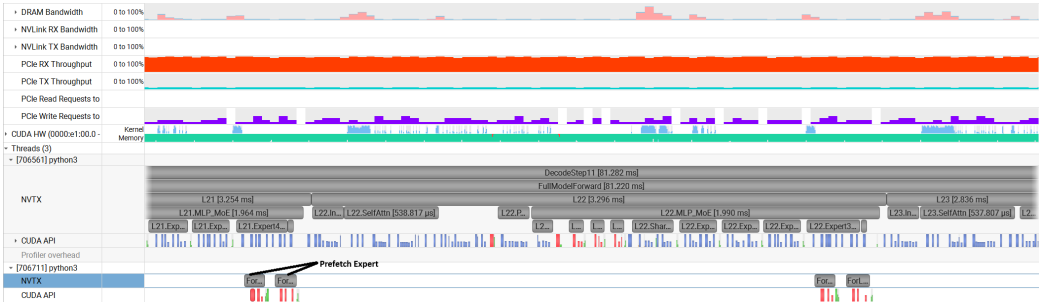
We next consider Qwen3.5-MoE in the same per-layer view, shown in Figure 8. This model routes each token to K=8 experts from a pool of 256 experts. Compared with Qwen1.5-MoE, its routed experts are smaller, so both expert transfer time and expert compute time are shorter. This makes the overlap window tighter and therefore makes Qwen3.5-MoE a harder case for prefetching. Nevertheless, the same bottleneck remains visible in the traces: expert computation is still shorter than PCIe transfer, so on-demand fetching continues to expose PCIe latency on the critical path.

Figure 8a shows the JIT baseline. As in the earlier model, the non-MoE portion of the layer, including self-attention and the surrounding normalization/post-attention regions, executes while PCIe RX is mostly idle. Thus, the communication window provided by the non-MoE part of the layer is left unused because the top-K experts are not yet known. Once the MoE block starts, PCIe RX becomes active and expert weights are fetched on demand. In this profiled case, one of the selected experts is already resident on GPU from the previous decoding step, so JIT still needs to transfer seven experts. The kernel activity shows that expert computations are short, whereas the memory-copy activity persists longer, leaving visible gaps where the GPU waits for the next expert weights. Even though Qwen3.5-MoE has smaller experts, the critical-path behavior is unchanged: transfer is still late, and GPU stalls still occur.

Figure 8b shows full CEPA on the same layer. The background thread, visible in the lower NVTX range, first issues the advance E_A transfer and later, once Signal E_B becomes available, issues the remaining prefetches. These two stages are visible in the figure and correspond exactly to CEPA's two-stage schedule: the first prefetch burst comes from the early signal, and the second burst comes from the calibrated refinement signal. In total, CEPA prefetched six experts, of which five were



(a) Annotated JIT fetching profile for one Qwen1.5-MoE layer. PCIe RX red bars show expert transfers, CUDA HW memory regions show transfer kernels, CUDA HW kernel activity shows expert computation, and NVTX ranges mark normalization, self-attention, MoE, expert transfer calls, and expert computation calls.



(b) Diagnostic two-candidate E_A-ranking ablation (not the reported CEPA configuration, which transfers one E_A candidate). One prediction is correct and one is wrong, so three routed experts remain on demand, but one useful transfer leaves the critical path.



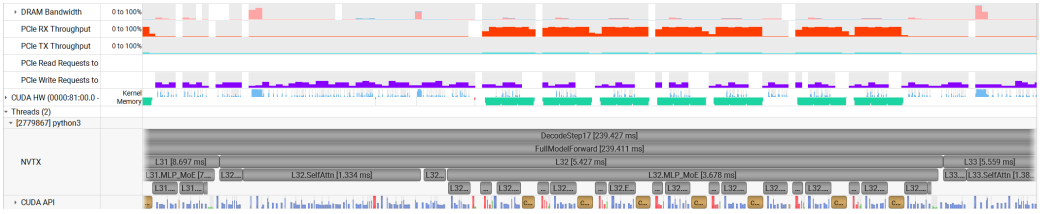
(c) Full CEPA. CEPA waits for E_B and then transfers calibrated experts. Two correct routed experts are prefetched before the MoE block, so only two routed experts remain to be fetched on demand.

Figure 7: Per-layer Nsight Systems profiles for Qwen1.5-MoE. Red PCIe RX bars show host-to-device expert transfers, CUDA HW kernel activity shows GPU computation, CUDA HW memory activity shows asynchronous memory copies, and NVTX marks the layer, attention, MoE, shared expert, routed experts, and background prefetch thread. CEPA moves expert transfers into otherwise idle PCIe windows, reducing GPU stalls caused by on-demand expert fetching.

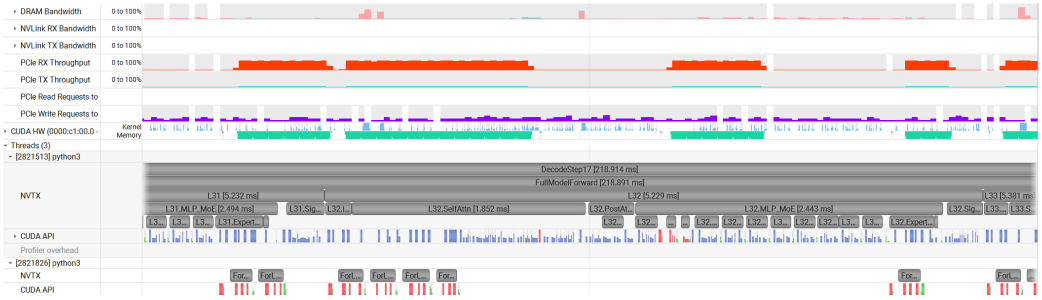
correct. Since one of the finally selected experts was already resident on GPU from the previous decode step, only two selected experts remained to be fetched on demand inside the MoE block.

This difference is the key result of the figure. Relative to JIT, CEPA moves the majority of useful expert transfers out of the MoE critical path and into the earlier non-MoE window. The background thread ensures that transfer orchestration does not stall the main CPU thread, while the two GPU streams allow expert transfers and model computation to proceed concurrently. As a result, the two remaining on-demand transfers in CEPA are fully overlapped with the computation of already-available experts, so their latency is effectively hidden. This is why the MoE region in the CEPA trace is shorter even though CEPA incurs extra prediction work.

An important point in this model is that the non-MoE window is large enough to accommodate even more prefetched experts, but in this profiled layer six prefetches are already sufficient to remove the exposed PCIe stalls. In other words, CEPA does not need to exhaust the full available overlap



(a) Qwen3.5-MoE with JIT fetching. PCIe RX is mostly idle during the non-MoE part of the layer and becomes active only inside the MoE block. One selected expert is already resident, so seven experts are fetched on demand.



(b) Qwen3.5-MoE with CEPA. The background thread issues one advance E_A transfer followed by the E_B refinement transfers. Five of the six total prefetched experts are correct; with one selected expert already resident, only two selected experts remain for on-demand transfer, and these are hidden by available expert computation.

Figure 8: Per-layer Nsight Systems profiles for Qwen3.5-MoE. CEPA shifts useful PCIe transfers into the non-MoE window and reduces critical-path expert fetching from seven experts to two, with the remaining transfers hidden behind available expert computation.

opportunity; it only needs to prefetch enough useful experts so that the remaining on-demand transfers can be hidden behind available expert computation. This is exactly what the figure shows: CEPA reduces the exposed transfer burden from seven experts in JIT to only two experts, and those two transfers are themselves hidden by concurrent expert execution.

Qwen3.5-MoE therefore illustrates an important point about CEPA. Because its experts are smaller, the margin for overlap is tighter than in Qwen1.5-MoE, so the absolute speedup is smaller. Even so, the figure shows that CEPA still uses the otherwise idle non-MoE communication window effectively, reduces GPU waiting inside the MoE block, and improves latency. In the target regime where GPU memory can hold the active computation state but not the full expert pool, CEPA remains beneficial even for small-expert models because it systematically shifts useful PCIe traffic earlier while preserving exact routed MoE execution.

As shown by the decode-time measurements in the Qwen3.5-MoE profiling figures, CEPA reduces the per-decode-step latency from 239.42 ms to 218.91 ms. CEPA prefetches six experts before the MoE block, five of which are correct; with one selected expert already resident, only two selected experts remain for on-demand transfer, and their latency is hidden behind available expert computation.