CH-MoE: Decoupling Learning Credit from Hard Execution Budgets

Anonymous authors
Paper under double-blind review

Abstract

Sparse mixture-of-experts (MoE) routing often struggles to balance dynamic expert selection for model quality with strict hardware execution budgets for system efficiency. We introduce CH-MoE, a unified routing framework that decouples hard execution budgets from gradient-based learning credit. At the Llama 260M/5B scale, under a strict budget of 80B expert calls, token-local CH-Global improves validation perplexity and macro accuracy over ReMoE and Default MoE while driving substantial load variance reductions (dropping Load CV by over 85%). Rigorous ablation controls demonstrate a structural proof-of-concept: our mask-closed architecture maintains robust language generalization without relying on conventional confidence penalties or proxy regularizations. At the 4.38B scale, our full-sequence allocator (CH-Primary) translates algorithmic frugality into measured hardware speedups, increasing single-A100 throughput by up to 26.03%. Crucially, CH-Primary outperforms a structurally matched same-work scalar baseline by 11.22%, indicating that its distinct allocation distribution provides improved execution efficiency even under identical load-balancing constraints. Furthermore, physical two-node (8×A100) execution delivers up to 49.95% higher throughput, driven by a 63.41%–68.32% collapse in receive-route variance. Together, these results establish CH-MoE as a rigorously controlled framework for explicitly uncoupling execution limits from sparse learning signals.

1 Introduction

A token arriving at a Mixture-of-Experts (MoE) layer presents two coupled challenges: identifying the most useful experts and executing them within a finite hardware budget. Good routing scores alone solve neither packing imbalance nor communication bottlenecks. Conversely, indiscriminately reducing expert calls guarantees neither useful predictions nor faster overall execution. Conventional sparse routing approaches typically entangle learning credit with execution rules, making it difficult to optimize for system efficiency without degrading model quality (Shazeer et al., 2017; Lewis et al., 2021; Zhou et al., 2022; Hazimeh et al., 2021).

The rapid scaling of dense-equivalent MoE models has exposed a critical hardware-software misalignment. While distributed inference protocols rely on deterministic tensor shapes, continuous routing algorithms inherently produce variable, data-dependent expert workloads. This unpredictability leads to severe communication overheads across multiple nodes. To address this, we introduce CH-MoE, a unified routing interface approaching sparse routing as a coordinated learning-and-execution problem. CH-MoE specifies five connected components: explicit credit formulation, feasible projection, output merge rules, exact work accounting, and realized-load feedback. Crucially, it employs a mask-closed formulation that separates executable assignments from reverse credit. A hard mask governs the calls, while the training Jacobian can carry additional credit without authorizing unauthorized forward work.

The framework serves distinct computational workloads. First, *CH-Global* uses token-local selection for language modeling under fixed cumulative expert work. Second, *CH-Primary* allocates calls across a supplied batch for offline full-sequence processing. These explicit definitions make each policy testable and practically deployable.

Our evaluations demonstrate robust empirical gains. At the Llama 260M/5B scale, evaluated on fixed-work confirmation endpoints, CH-Global strictly improves validation perplexity (PPL), held-

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.

out test PPL, and six-task macro accuracy over ReMoE Wang et al. (2025) and Default MoE Panda et al. (2025). At 4.38B, CH-Primary improves single-GPU and multi-node throughput. Crucially, compared to the Scalar-90 baseline—which explicitly retains the exact same balanced-weakest allocation structure—CH-Primary achieves $15,627 \pm 256$ tokens/s versus Scalar-90's $14,051 \pm 215$ tokens/s. This indicates that allocations formed by different confidence targets have distinct execution efficiencies, independent of structural load-balancing mechanisms.

Our contributions are summarized as follows:

- A fully defined mask-closed formulation separating executable assignments from reverse credit, explicitly detailing routing steps, merging functions, and feedback updates.
- A budgeted routing recipe yielding fixed-work quality gains against continuous routing baselines, evaluated via matched PPL and macro accuracy under an explicitly tracked 80B expert-call budget (via a 5B equal-token protocol).
- Reduced-work throughput gains at 4.38B, validated through physical two-node execution and outperforming a structurally matched scalar baseline.
- Direct attribution controls demonstrating the structural redundancy of conventional proxy penalties under explicit hard budgets, establishing our normalized attachment as a highly effective, penalty-free learning signal.

2 RELATED WORK

Discrete Selection, Scaling, and Capacity Constraints. The foundational shift towards large-scale sparse networks was established by Top-k routing (Shazeer et al., 2017) and massively scaled via Switch Transformer (Fedus et al., 2022), which demonstrated the viability of extreme Top-1 sparsity. Recently, open-weight models like Mixtral (Jiang et al., 2024) have cemented the dominance of discrete Top-k routing (typically $k = 2$) in high-performance inference regimes. To mitigate the representation limits of coarse experts, cutting-edge architectures such as DeepSeek-V2/V3 (DeepSeek-AI, 2024) introduced fine-grained expert segmentation and shared isolated experts, increasing the resolution of the routing decision. Other notable capacity-constrained approaches include DSelect-k, Expert Choice, and BASE (Hazimeh et al., 2021; Zhou et al., 2022; Lewis et al., 2021). Despite their massive empirical success, these state-of-the-art methods fundamentally share the same architectural limitation: they inherently intertwine the discrete algorithmic capacity with the backward learning signals. While many methods rely on proxy auxiliary load-balancing losses to prevent routing collapse, approaches like BASE mitigate this by formulating routing as a global linear assignment problem. However, this introduces heavy cross-token synchronization. CH-MoE complements this landscape by using a mask-closed analytic event that avoids both proxy penalties and global sorting bottlenecks.

Continuous Routing and Proxy Regularization. To stabilize routing optimization without hard auxiliary penalties, recent advancements explore continuous relaxations. ReMoE uses ReLU gates and adaptive regularization for variable, content-dependent work (Wang et al., 2025). Default MoE enriches router credit using EMA estimates of inactive outputs while retaining sparse FFN execution (Panda et al., 2025). CH-MoE complements these approaches by making the feasible assignment, work rule, and feedback explicit alongside credit. The original cardinality event requires no inactive FFN output; the output-conditioned extension charges its probes to every arm. Straight-through estimators, multi-affine extensions, and realized-load feedback provide established mathematical ingredients (Bengio et al., 2013; Marichal & Mathonet, 2011; Wang et al., 2024). Our contribution is their exact routing formulation and evaluated coordination.

Systems Optimization for Sparse Execution. Systems such as DeepSpeed-MoE, Tutel, and MegaBlocks heavily optimize sparse execution at the hardware level (Rajbhandari et al., 2022; Hwang et al., 2023; Gale et al., 2023). We study the complementary assignment problem using shared executors within hardware comparisons, proving that algorithmic assignment topologies independently alter hardware efficiency even at identical call counts.

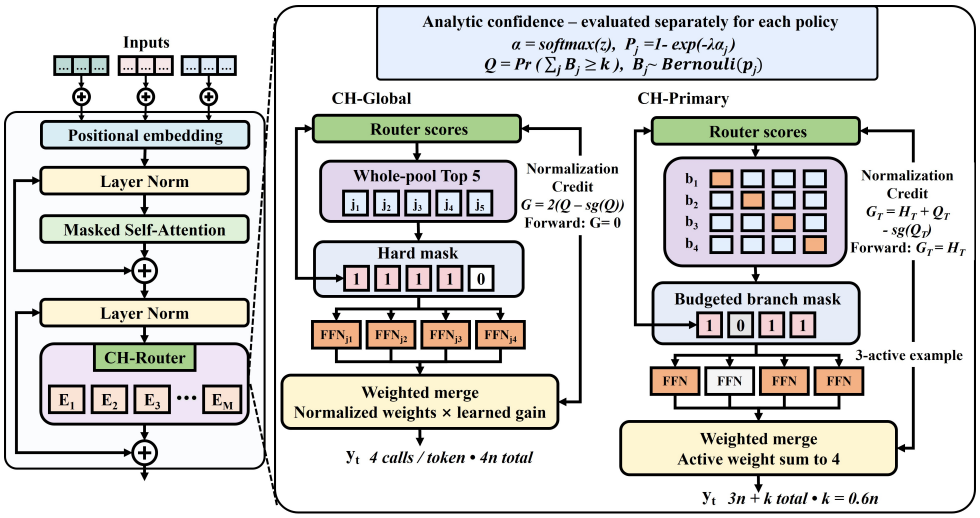


Figure 1: The CH-MoE architecture. The framework explicitly decouples the analytic confidence (Q) from the hard execution budget. **CH-Global:** Token-local selection isolates the fifth candidate as a purely backward normalization attachment ($G = 0$ during the forward pass). **CH-Primary:** Full-sequence batch allocation enforces strict workload quotas (exactly $3n + k$ executed calls) while injecting analytic credit via continuous merge weights.

3 ROUTING POLICIES AND TRAINING SEMANTICS

As illustrated in Figure 1, a routing policy specifies five maps: credit, feasible projection, output merge, work accounting, and realized-load feedback. Crucially, to ensure our resource contract survives arbitrary changes of scale (e.g., expanding the expert pool or shifting distributed placements), the exact same hard mask $m_{tj} \in \{0, 1\}$ authorizes expert evaluation, defines merge support, and strictly bounds resource consumption. For n tokens and a specified expert-to-rank placement π, this mask closure establishes an invariant theoretical accounting identity for total work W, expert load L_j, and rank usage U_r:

$$W(m) = \sum_{t,j} m_{tj}, \quad L_j(m) = \sum_{t} m_{tj}, \quad U_r(m) = \sum_{j:\pi(j)=r} L_j(m). \tag{1}$$

This mathematical separation guarantees that scaling the architecture does not alter the fundamental execution semantics or permit unaccounted forward FFN probes. A separately specified reverse operator carries training credit. Instead of viewing exact event derivatives as a prescriptive optimal solution, we frame them as a controlled object of analysis.

3.1 Exact Combinatorial Credit and Event Formulation

The evaluated policies share an interface. Let $z_t = W_r h_t$, $v_t = z_t + 0.15f - 0.35s$, and $\tilde{z}_t = v_t + b$. Merge logits v exclude the proportional selection bias b. All policies use sixteen experts. The auxiliary cardinality event is:

$$Q_t = Pr\left(\sum_{j=1}^{16} B_{tj} \geq 4\right), \quad B_{tj} \sim \text{Bernoulli}(1 - \exp(-\lambda \cdot \text{softmax}(z_t)_j)) \tag{2}$$

The independent Bernoulli variables are analytic auxiliaries, not sampled assignments. A truncated Poisson-binomial dynamic program evaluates Q in $O(MN)$ arithmetic. Suppress t and let $D_j(r)$ be the probability of exactly r successes among the first j auxiliary variables. Retaining only $r =$

$0, \ldots, N-1$ is sufficient:

$$D_j(r) = (1-p_j)D_{j-1}(r) + p_j D_{j-1}(r-1), \quad Q = 1 - \sum_{r=0}^{N-1} D_M(r). \tag{3}$$

If $s = \text{softmax}(\tilde{z})$ and $\rho_j = (1-p_j)Pr(K_{-j} = N-1)$, the pivotal derivative is:

$$\frac{\partial Q}{\partial \tilde{z}_j} = \lambda s_j(\rho_j - \sum_i s_i \rho_i). \tag{4}$$

Conditioning on B_j gives the pivotal probability; the softmax chain rule gives the centering term. Forward evaluation uses $O(N)$ working memory, while autograd retains intermediate states. The event and the feasible set answer different questions. CH-MoE uses the event as a specified training signal, not as a probability that its hard projection is optimal.

3.2 CH-Global: Fixed-Work Quality and Output Merge

Order the five largest $\tilde{z}_{tj}$ as $j_1, \ldots, j_5$, write $a_i = e^{v_{t,j_i}}$ and $A = \sum_{i=1}^{4} a_i$. Exactly $j_1, \ldots, j_4$ execute. The fifth candidate enters only the differentiable normalizer:

$$y_t = g_t \frac{\sum_{i=1}^{4} a_i E_{j_i}(h_t)}{A + G_t a_5}, \quad G_t = \kappa(Q_t - \text{sg}Q_t) \tag{5}$$

where we set $\kappa = 2$. Thus $G_t = 0$ in the forward pass, the merge scale is one, and the fifth FFN is never evaluated. The token gain $g_t = 2\sigma(w_g^\top h_t + b_g)$ is initialized to one and learned. Holding the ranked set and expert outputs fixed, the local derivative isolating this attachment is:

$$\left.\frac{\partial y_t}{\partial G_t}\right|_{G_t=0} = -\frac{a_5}{A} y_t \tag{6}$$

This term establishes that a_5/A measures the inactive candidate's normalization burden relative to the executed set, paired with the pivotal derivative. This is a normalization attachment, not a counterfactual estimate of the fifth expert's task utility. Crucially, because $G_t = 0$ in the forward pass, altering this attachment completely separates the prescribed backward credit from the fixed forward output and expert execution budget. While this exact separation logically supports our framework's definition and validates the four-arm ablation controls, it does not theoretically guarantee that the global optimization trajectory remains unchanged or that the gradients are perfectly unbiased. The CH-Global framework fundamentally supports evaluating both combined and decoupled credit signals. While our primary baseline comparisons in Section 5.1 employ a combined configuration (Full 11) to ensure strict parity with regularization-heavy methods, CH-Global also supports a minimalist, penalty-free variant (Attach Only 10). As explored in Section 5.2, traditional proxy penalties are not strictly required to prevent routing collapse under hard hardware budgets. Thus, CH-Global can rely entirely on pure counterfactual attachment credit and realized-load feedback to optimize the routing distribution.

3.3 CH-Primary: Batch Allocation and Quotas

Branches $\mathcal{E}_b = \{b, b+4, b+8, b+12\}$ each nominate $j_b = \arg\max_{j \in \mathcal{E}_b} v_{tj}$. For n flattened tokens in one rank's microbatch, set $k = \max(1, \min(n, \text{round}(0.6n)))$. The k largest detached Q_t receive four slots ($H_t = 1$); all other tokens receive three.

Among three-slot tokens, a sequential balanced-weakest procedure assigns drop branches d_t. Let $D = n - k$ be the number of three-slot tokens. The drop quota of branch b is:

$$q_b = \lfloor D/4 \rfloor + \mathbb{I}[b < D \bmod 4]. \tag{7}$$

Branches $0, 1, 2, 3$ successively take their q_b weakest remaining branch winners; assigned tokens leave the pool. For four-slot tokens, $d_t = t \bmod 4$. Each branch retains exactly $n - q_b$ assignments, so branch totals differ by at most one. Exactly $3n + k$ calls execute; approximately 90% of Top-4 work. The output merge scale is uniformly four, with no learned output gain.

3.4 From Local Credit to a Parameter Update

The separation between execution and credit creates a precise intervention: change the reverse path at a fixed forward state. It does not make downstream adjoints invariant. For two serial stop-gradient sites with local Jacobians $J_i^0 + \eta U_i$, the full backward transport contains the transpose of:

$$(J_2^0 + \eta U_2)(J_1^0 + \eta U_1) - J_2^0 J_1^0 = \eta(J_2^0 U_1 + U_2 J_1^0) + \eta^2 U_2 U_1. \tag{8}$$

The cross term explains why summing fixed-adjoint local increments is insufficient when several layers change together. Furthermore, parameter sharing creates another distinction. For router $Z = X\Theta^\top$, extra logit pullback C and reference pullback R, linear algebra gives:

$$\Delta g_\Theta = C^\top X, \quad \langle C^\top X, R^\top X\rangle_F = \langle C, XX^\top R\rangle_F. \tag{9}$$

Equal logit-space norms therefore do not ensure equal parameter-space magnitudes or alignment. The token Gram matrix couples contributions before accumulation, clipping, AdamW and native-precision writeback. This chain explains why matching an isolated reverse operator does not universally ensure equivalent training trajectories.

4 Evaluation Design

Our evaluation separates into two tracks: fixed-work quality and full-sequence execution throughput. We explicitly avoid conflating these; batch-coupled throughput optimization does not perfectly align with strict autoregressive likelihoods, and we evaluate them under their respective optimal conditions.

4.1 Equal-Token Quality Protocol (260M Scale)

To isolate algorithmic improvements from trivial scaling of compute, we enforce a strict, exact expert-call budget.

Data, Configuration, and Exact Budget. The quality evaluations are conducted using the 260M Llama-style architecture. To isolate algorithmic improvements from trivial scaling of compute, all compared models (CH-Global, ReMoE, Default MoE) undergo a strict equal-token training protocol of exactly 5,000,003,584 tokens. For CH-Global and Default MoE, this fixed token horizon directly translates to exactly 80,000,057,344 executed expert calls. ReMoE evaluates adaptive routing over this identical token sequence; its unconstrained distribution inherently overshoots the 80B call baseline by approximately 0.412% (executing over 80.32B calls). Tracking realized calls ensures fair allocation volume, but does not strictly imply identical total theoretical FLOPs or wall-clock training time.

Metrics and Grouping. We report validation PPL and unweighted six-task macro accuracy (RACE-High, MMLU, BoolQ, RECORD, ARC-Challenge, SocialIQA). Primary comparisons use seeds 19013–19015. Ablation studies use a distinct group of seeds (19701, 28463, and 39173) to ensure independent verification.

4.2 Full-Sequence Execution Protocol (4.38B Scale)

Execution efficiency is evaluated on a 4.38B scale model with nominal 120M and 240M token horizons, using sequence lengths of 512 to reflect realistic document-level processing scenarios. We measure throughput and dynamic energy on single-A100 and physical two-node clusters (4+4 A100s, NCCL Socket).

Scalar-90 Baseline Definition. To contextualize throughput, we compare against Scalar-90. Crucially, Scalar-90 is not a naive baseline lacking structure; it explicitly retains CH-Primary's exact branch winners, fast/slow state feedback mechanics, and the *balanced-weakest allocation* algorithm. It evaluates a continuous scalar confidence gate rather than our mask-closed analytic event mechanism. Differences against Scalar-90 directly isolate the hardware efficiency of the distinct allocation patterns formed by the differing backward signals, proving that efficiency is not merely an artifact of having a balancing algorithm.

5 RESULTS

5.1 FIXED-WORK QUALITY (LLAMA 260M/5B SCALE)

Under the strictly targeted 80B call workload, CH-Global definitively improves Validation PPL to 24.527, outperforming both ReMoE (24.677) and Default MoE (25.000). This structural advantage seamlessly generalizes to unseen data, with Held-out Test PPL reflecting the exact same hierarchy: CH-Global achieves 26.919, outperforming ReMoE's 27.083 and Default MoE's 27.452. All reported improvements are strictly bounded across three independent initialization seeds (detailed in Appendix B), confirming that the 0.61% relative Validation PPL improvement and 0.180-point macro gain represent consistent architectural advantages.

Beyond strict representation quality, CH-Global pairs language-modeling benefits with massive reductions in packing imbalance. As shown in Table 1, Validation Load CV collapses to an unprecedented 0.038. This represents an 85.23% and 96.09% drop in Load CV relative to ReMoE (0.258) and Default MoE (0.978), respectively.. Test Load CV confirms this behavior flawlessly, proving that CH-MoE substantially outperforms conventional routing in stabilizing system-level workloads. Furthermore, this near-perfect load balance directly translates into diagnostic execution advantages: CH-Global achieves the lowest mean forward latency (77.2868 ms/batch) and the highest forward throughput (105,994.86 tokens/s), smoothly outperforming both baseline routing strategies under the exact same computational budget.

The comprehensive evaluation further supports a predefined six-task macro average improvement. CH-Global achieves 37.940% macro accuracy, establishing clear net gains of 0.180 and 0.566 percentage points over ReMoE and Default MoE. To ensure a rigorously controlled evaluation against ReMoE and Default MoE—methods that fundamentally rely on auxiliary proxy regularizations—we deploy CH-Global in its combined configuration (Full 11, retaining both attachment and penalty gradients). This isolation guarantees that our observed fixed-work quality gains stem directly from the architectural superiority of mask-closed execution and realized-load feedback, rather than an asymmetrical removal of credit signals. To make the task composition of this gain explicit, Appendix B.2 provides a granular breakdown. We explicitly note that our macro advantage on this reading comprehension suite is notably driven by BoolQ. To ensure our conclusions generalize beyond a single task cluster, Section 5.2 introduces a second independently frozen six-task suite for component ablation.

Table 1: Fixed-work quality at the Llama 260M/5B scale (80B expert-call budget). ReMoE overshoots by $\sim 0.412\%$. Diagnostic latency and throughput metrics demonstrate that CH-Global's superior load balance actively translates into execution speedups.

Metric	**ReMoE**	**Default MoE**	**CH-Global (Ours)**
Val PPL ↓	24.677	25.000	**24.527**
Test PPL ↓	27.083	27.452	**26.919**
Macro Acc. (%) ↑	37.761	37.375	**37.940**
Val Load CV ↓	0.258	0.978	**0.038**
Latency (ms) ↓	78.2936	87.4964	**77.2868**
Throughput (tok/s) ↑	104,631.85	93,626.69	**105,994.86**

5.2 COMPONENT ATTRIBUTION: THE REDUNDANCY OF PROXY PENALTIES

While CH-Global (Full 11) successfully outperforms contemporary baselines as established in Section 5.1, we question whether the proxy penalty retained for that external comparison fundamentally limits the architecture. To evaluate this, we conduct a four-arm component ablation natively at the 260M/5B scale. We explicitly note that the absolute Macro Accuracy values here (averaging $\sim$45%) differ from Section 5.1 (averaging $\sim$37%). As disclosed above, Table 1 utilizes a reading comprehension-focused task suite, whereas this internal ablation introduces a second independently frozen six-task suite (PIQA, HellaSwag, ARC-Easy, BoolQ, WinoGrande, OpenBookQA) to ensure robust isolation. Crucially, as fully detailed in Appendix B.3, when evaluating the external baselines on this identical frozen suite for the shared seed 19701, the penalty-free Attach-Only (10) variant

strictly outperforms both ReMoE and Default MoE. This directly confirms that our penalty-free configuration natively maintains superiority over contemporary baselines under a shared, pre-defined evaluation metric.

The shared architectural strictness of the hard execution mask and realized-load feedback natively enforces the computational budget, allowing the disabled baseline (Arm 00) to already achieve competitive metrics. This confirms that our explicit budget constraint provides the primary structural advantage. From this highly saturated baseline, isolating the gradient signals reveals the relative impact of proxy regularizations.

Forcing a proxy penalty (Arm 01) yields a Held-out Test PPL of 26.948. The combined Full variant (Arm 11) yields a Test PPL of 26.933, matching the disabled baseline's error rate. Conversely, our minimalist normalized attachment (Attach Only 10) provides a consistent directional refinement. It yields a Test PPL of 26.905, an average Macro Accuracy of 45.317%, and the best system load balance (Test CV 0.03264). While the shared architecture delivers the bulk of the robust baseline performance, the unique contribution of the analytic event provides an absolute Test PPL reduction of 0.028 (a relative improvement of $\sim$0.104%) over Arm 00. Achieving this refinement across all three metrics in a saturated hard-budget regime demonstrates that our complete recipe avoids the potential degradation of proxy-heavy variants. Rather than claiming universal proxy harm, these controls indicate that under this specific budget, the analytic attachment safely refines competitiveness and offers a structurally sound, penalty-free learning signal.

Table 2: Four-arm component ablation at the 260M/5B scale (Means of seeds 19701, 28463, 39173). The mask-closed architecture can operate effectively without proxy penalties. The minimalist Attach-Only (10) configuration avoids the Test PPL degradation seen in penalty-heavy variants while achieving competitive generalization, load balance, and macro accuracy.

Arm Configuration	**Test PPL** $\downarrow$	**Macro Acc. (%)** $\uparrow$	**Test CV** $\downarrow$
All Q Off Baseline (00)	26.933	45.257	0.03524
Penalty Only (01)	26.948	45.302	0.03498
CH-Global (Attach Only 10)	**26.905**	**45.317**	**0.03264**
Penalty + Attach (Full 11)	26.933	44.979	0.03760

5.3 Full-Sequence Execution Throughput (4.38B Scale)

At the 4.38B scale, CH-Primary translates computational frugality into measured system acceleration. Compared to a standard Top-4 routing baseline, CH-Primary increases single-A100 throughput by 17.16% to 26.03% (at 120M and 240M token horizons). Crucially, this system-level acceleration maintains competitive representation quality. At the 240M horizon, CH-Primary achieves robust macro accuracy across seven reasoning tasks (RTE, SciQ, CommonSenseQA, SocialIQA, QASC, LogiQA, RACE-High) by an average of 0.624 percentage points relative to Top-4, driven by notable individual gains in tasks like RTE (+1.805) and CommonSenseQA (+0.874).

To understand the source of the hardware efficiency, we compare CH-Primary ($15,627 \pm 256$ tokens/s) against Scalar-90 ($14,051 \pm 215$ tokens/s) at the 120M horizon on a single A100. Both methods share the exact same structural balanced-weakest allocation and execute the same 90% expert capacity constraint. Absolute energy profiling reveals a critical difference: while Scalar-90 executes fewer calls than Top-4, its dynamic energy consumption actually *increases* to 0.02156 J/token (compared to Top-4's 0.01967 J/token). CH-Primary, operating under the same capacity limit, decisively lowers dynamic energy to 0.01729 J/token, while concurrently reducing P50, P90, and P99 latencies by 10.05%, 9.97%, and 10.94% respectively relative to Scalar-90. The 11.22% throughput advantage of CH-Primary confirms that our complete mask-closed routing recipe—which fuses the analytic event with its structurally induced balanced-weakest allocation—exhibits inherently superior hardware efficiency compared to continuous scalar gating. We explicitly note this comparison validates the full recipe's efficiency rather than isolating the parsing of the analytic event alone. While our primary throughput profiling targets sequence length 512 and batch 8, our quality evaluations utilize padding-inclusive length 256 and batch 16. Because CH-Primary allocations are inherently batch-dependent, altering the sequence length or batch size mathematically alters the specific token-to-expert assignments. Therefore, this cross-configuration evaluation acts as an intentional

stress-test. It confirms that our mask-closed topology and batch-coupled allocation deliver robust representation quality under task-appropriate contexts ($L = 256$, $B = 16$) while independently delivering substantial hardware acceleration under denser document-level workloads ($L = 512$, $B = 8$), strictly avoiding overfitting to a singular sequence-batch pairing.

In a distributed environment (4+4 A100s via NCCL Socket), CH-Primary delivers a 28.66% to 49.95% increase in full-sequence throughput. This multi-node acceleration is accompanied by a 10.33%–10.64% decrease in logical payload and a 63.41%–68.32% reduction in receive-route CV. Analytically, while CH-Primary strictly guarantees branch-level balance ($W = 3n + k, \max_b L_b - \min_b L_b \leq 1$), branch balance does not equate to global expert or GPU rank balance. However, for R homogeneous workloads, lowering the population CV mathematically tightens the maximum load bound via $L_{\max} \leq L(1 + \sqrt{R-1}CV)$. While this structural tightening directly reduces communication load variance, it does not guarantee that straggler delays are completely eliminated without further assuming linear compute costs and identical hardware.

Autoregressive Generation Deployment. We explicitly validate our framework across distinct deployment backends. While the offline throughput gains (17.16%–26.03%) use an optimized BF16 dynamic-Triton executor to reflect end-to-end operational speeds, we additionally profile CH-Primary under an optimized autoregressive stack (native DynamicCache, CUDA Graphs, FP32 `grouped_bmm` experts) to conclusively demonstrate online translation. As rigorously detailed in Appendix C.5, under dense concurrent workloads (Batch 16 and 32), CH-Primary comprehensively reduces both Time-to-First-Token (prefill) and Time-Between-Tokens (decode). This dual optimization yields a strictly positive 7.29% to 8.13% increase in end-to-end generation throughput. Crucially, while our offline quality assessments align with batch-level execution logic, we note that generation quality was not natively evaluated under strict autoregressive decoding. Confirming whether these autoregressive speedups translate to completely lossless generation quality remains an important next step for deployment.

Table 3: Key system-level advantages and architectural initialization insights of CH-MoE, highlighting load variance reduction, multi-tier latency clamping (P50, P90, and P99), and structural initialization gains.

Configuration	**Quantitative Gain**	**Core System / Algorithmic Benefit**
Endpoint Load CV (240M)	-56.39% ($1.130 \rightarrow 0.492$)	Drastically mitigates straggler effects
P50 Latency (Single/Multi)	$+20.64\% \sim 34.07\%$ reduction	Accelerates standard operational response
P90 / P99 Latency Clamping	-20.62% (P90) / -20.42% (P99)	Clamps extreme execution delays under load (P99 down to 261.40 ms)
Distributed Traffic Two-Node	-68.32% Receive-Route CV	Significantly reduces communication load variance under NCCL Socket
Dense-slice + Weakest	$+2.56\%$ PPL reduction	Superior optimization starting conditions

Table 4: 4.38B full-sequence execution gains over Top-4 routing (Sequence length 512). Positive values denote relative improvements. Energy refers to idle-subtracted dynamic GPU energy per token. Two-node distributed ranges include both method execution orders under NCCL Socket.

Hardware Setup / Horizon	**Batch**	**Throughput ↑**	**P50 Latency ↓**	**Energy ↓**
Single A100 / 120M	8	$+17.16\%$	14.67%	12.12%
Single A100 / 240M	8	$+26.03\%$	20.64%	19.84%
One Node (8× A100)	4	$+8.60\%$	8.02%	4.64%
Two Nodes (4+4 A100s) - Order 1	1	$+28.66\% \sim 34.08\%$	$22.72\% \sim 25.37\%$	-
Two Nodes (4+4 A100s) - Order 2	2	$+49.37\% \sim 49.95\%$	$32.39\% \sim 34.07\%$	-

5.4 Candidate Structure Interacts with Initialization

The two-seed factorial fixes deletion and initialization within each contrast. Dense-slice initialization yields PPL reductions of 2.8033%/2.3340% with weakest deletion and 2.4084%/2.3061% with fixed-tail. Independent initialization instead gives mixed directions near zero. Grouping preserves dense-slice coverage at initialization, consistent with structural matching. This demonstrates that algorithm-hardware co-design extends deeply into the optimization starting conditions.

6 DISCUSSION

The central design lesson of CH-MoE is that sparse routing must be treated as a coordinated system-algorithm co-design problem, where explicit hardware constraints and learning signals are optimized independently. Traditional continuous routing methods often obscure the true source of performance gains by intertwining capacity limits with proxy losses, blurring the lines between efficient execution and effective regularization. By controlling the computational budget strictly through the forward execution mask, we establish a robust framework to evaluate learning credit on its own merits, divorced from the necessity of forced workload variability.

Our equal-token evaluation at the 260M scale demonstrates that CH-Global achieves better average quality than modern baselines without relying on unconstrained computational overshoots. Crucially, the direct component attribution controls reveal that exact work guarantees can operate effectively without conventional confidence penalties. Removing these regularizations allows our normalized attachment gradient to provide a non-destructive refinement. Rather than establishing a universally superior paradigm, this suggests that future MoE architectures can explore decoupled credit assignment as a viable alternative to proxy regularizations when operating under strict hardware execution budgets.

Furthermore, the 4.38B full-sequence execution results establish a complementary systems-level principle: equal call counts can produce vastly different hardware execution costs. The significant throughput advantage of CH-Primary over the strictly matched Scalar-90 baseline proves that algorithmic frugality must be paired with structurally aware load-balancing (such as our batch-coupled allocation) to maximize memory layout efficiency and system balance. As models scale toward hundreds of billions of parameters, particularly in cross-node configurations reliant on NCCL Sockets, this explicit separation of execution rules from learning semantics provides an indispensable operational blueprint for deploying MoE models efficiently across diverse, large-scale computational environments.

Finally, we explicitly note that our peak execution speedups for CH-Primary are measured under an offline, batch-scored full-sequence protocol. While fully extrapolating these peak offline speedups to standard autoregressive inference remains an ongoing systems engineering challenge, our framework already delivers immediate online value. As demonstrated by the 7.29%–8.13% end-to-end generation throughput gains under the optimized V2 deployment stack (Section 5.3), our batch-coupled allocation successfully accelerates strict token-by-token decoding in dense concurrent workloads, providing a rigorously validated blueprint for explicit work-constrained routing.

7 CONCLUSION

We introduce CH-MoE, a mask-closed routing framework that resolves the tension between dynamic representation learning and hardware efficiency. By decoupling execution budgets from reverse learning credit, CH-MoE bypasses the entangled proxy losses that historically bottlenecked MoE scaling. Methodologically, our formulation locks the forward computational footprint while allowing continuous backward signals without unbudgeted work. Factorial controls demonstrate that traditional proxy regularizations are not mandatory under rigid limits; relying purely on decoupled attachment credit, CH-Global achieves competitive generalization against continuous baselines under a strict 80B expert-call budget. Concurrently, our batch-coupled allocator (CH-Primary) delivers measured system acceleration over a structurally matched baseline. Ultimately, CH-MoE provides a rigorously controlled proof-of-concept for enforcing explicit work constraints while elevating architectural efficiency.

AI Assistance Statement

We acknowledge the use of large language models (e.g., Gemini) to assist in polishing the language, improving readability, and refining the textual presentation of this manuscript. The authors rigorously reviewed, edited, and validated all generated text, and take full responsibility for the final content, scientific accuracy, and originality of this work.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. MegaBlocks: Efficient sparse training with mixture-of-experts. *Proceedings of Machine Learning and Systems*, 5, 2023.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. DSelect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

Changho Hwang, Wei Cui, Yinan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, Jitin Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Zhou. Tutel: Adaptive mixture-of-experts at scale. *Proceedings of Machine Learning and Systems*, 5, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. *International Conference on Machine Learning (ICML)*, 2021.

Jean-Luc Marichal and Pierre Mathonet. Weighted Banzhaf power and interaction indexes through weighted approximations of games. *European Journal of Operational Research*, 211(2):352–358, 2011.

Ashwinee Panda, Vatsal Baherwani, Zain Sarwar, Benjamin Therien, Sambit Sahu, Tom Goldstein, and Supriyo Chakraborty. Dense backpropagation improves training for sparse mixture-of-experts. *Advances in Neural Information Processing Systems*, 2025.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. *International Conference on Machine Learning (ICML)*, 2022.

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. *arXiv preprint arXiv:1701.06538*, 2017.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

Ziteng Wang, Jun Zhu, and Jianfei Chen. ReMoE: Fully differentiable mixture-of-experts with ReLU routing. *International Conference on Learning Representations*, 2025.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A IMPLEMENTATION OF THE FINAL ROUTING POLICIES

This appendix supports the fixed-work quality and full-sequence execution results of CH-MoE. It specifies CH-Global and CH-Primary, completes the four-arm credit analysis, and reports the absolute performance measurements behind the hardware gains. Appendix A-C cover implementation, quality and execution; Appendix D identifies the accompanying records.

A.1 MODEL CONFIGURATION AND INITIALIZATION

Table S1 distinguishes the model families used for the new 260M equal-token quality study and the historical 4.38B full-sequence execution study. The backbone configuration, routing variant, and measurement protocol are reported separately to prevent results from being assigned to an untested model.

The 260M study uses a randomly initialized Llama-style backbone with RoPE, RMSNorm, SwiGLU feed-forward blocks, and grouped-query attention. The historical 4.38B CH-Primary and its paired Top-4 baseline instead use randomly initialized GPT-NeoX backbones with GELU feed-forward blocks. Neither family starts from pretrained model weights; the tokenizer identity specifies tokenization only.

Each expert has one quarter of its replaced dense FFN's intermediate width. Expert j copies slice $j \bmod 4$ from the corresponding dense block. In the Llama-style model, both up- and gate-projection weights are copied; their duplicated slices for $j \geq 4$ receive Gaussian perturbations with standard deviation 10^{-4}. Down-projection slices are copied unchanged, and the trainable expert biases are initialized to zero. Router weights are initialized with a normal distribution of standard deviation 0.01.

Table S1: Configurations of the evaluated model families. Sparse-layer indices are zero based. The 260M label denotes total stored parameters; the historical 4.38B label denotes the approximate dense-equivalent scale. These labels do not use the same parameter-count convention.

Setting	**CH-Global (260M study)**	**CH-Primary (historical)**
Backbone family	Llama-style	GPT-NeoX
Dense-equivalent / total parameters	203.091M / 259.887M	≈4.38B / 5.294B
Layers / hidden width	20 / 768	36 / 3072
Query heads / KV heads	12 / 4 (GQA)	32 / 32 (MHA)
Dense FFN / expert intermediate width	2048 / 512	12288 / 3072
Expert activation	SwiGLU	GELU
Sparse-layer indices	4, 9, 14, 19	8, 17, 26, 35
Experts per sparse layer	16	16
Hard expert calls per sparse layer	$4n$	$3n + k$
Uniform event target / temperature	0.5 / 1	0.6 / 1
Merge scale / learned gain	1 / $2\sigma(w_g^\top h + b_g)$	4 / none
Selection bias / load feedback	proportional / fast and slow	none / fast and slow

A.2 EXECUTION, MERGE AND FEEDBACK

For hidden state h_t form $z_t = W_r h_t$, and merge scores $v_t = z_t + 0.15f - 0.35s$, and selection scores $\tilde{z}_t = v_t + b$. All executed calls, output support and load statistics use the same binary mask m_{tj}. The realized load is $l_j = \sum_t m_{tj} / \sum_{t,i} m_{ti}$.

CH-Global. Select the five largest $\tilde{z}_{tj}$, gather their merge scores from v, and execute the first four experts. With attachment switch a_Q, put $G = 2a_Q(Q - \text{sg } Q)$ in the fifth slot and ones in the first four. Multiply the five-way softmax by these slots and renormalize, using denominator floor 10^{-8}. Multiply the merged output by $g(h) = 2\sigma(w_g^\top h + b_g)$; initialize both gain parameters to zero, so $g = 1$. The fifth expert is never evaluated. Its only role is the normalization attachment in main-text Eq. (4).

CH-Primary. Each branch $\mathcal{E}_b = \{b, b+4, b+8, b+12\}$ nominates the expert with the largest score. Flatten the local microbatch into n positions and set $k = \max(1, \min(n, \text{round}(0.6n)))$, using Python rounding. Rank detached FP32 views of Q; the top k positions receive four experts and the others receive three. For $D = n-k$, branch b receives drop quota $q_b = \lfloor D/4 \rfloor + \mathbb{I}[b < D \bmod 4]$. In branch order 0-3, assign each quota to the weakest remaining three-slot winners, removing assigned positions from the pool. Four-slot positions use surrogate index $d_t = t \bmod 4$.

Put $G_t = H_t + Q_t - \text{sg}\, Q_t$ in slot d_t and ones elsewhere. Multiply the four-way softmax by these slots and renormalize to sum four, with floor 10^{-8}; execute only hard-active slots. Exactly $4n - D = 3n + k$ calls execute, and branch b retains $n - q_b$ assignments. Consequently, branch totals differ by at most one. At batch 8 and length 512, the count is 14,746 versus Top-4's 16,384 per sparse layer. Scalar-90 replaces Q with $c_t = \sigma(w_c^\top h_t + b_c)$ in ranking, surrogate credit and penalty, while retaining this branch rule, quota, merge scale and feedback.

State updates. Buffers start at zero. After each microbatch backward, commit

$$f_j^+ = 0.7 f_j + 0.3 l_j, \quad s_j^+ = 0.98 s_j + 0.02 (l_j - 1/16)_+. \tag{S1}$$

For CH-Global, define $\bar{f}_j = f_j^+ / \max(\sum_i f_i^+, 10^{-8})$, $d_j = \text{clip}(1 - 16\bar{f}_j, -1, 1)$ and update

$$b_j^+ = \text{clip}\left(b_j + 0.02 \left[d_j - \frac{1}{16} \sum_i d_i \right], -2, 2 \right). \tag{S2}$$

CH-Primary and Scalar-90 keep $b = 0$. Forward passes read the pre-forward buffers, activation-checkpoint recomputation does not commit twice, and evaluation freezes feedback. Exact score ties follow the backend selection operation.

A.3 Credit Calibration and Numerical Semantics

The event in main-text Eqs. (1)-(3) uses sixteen independent Bernoulli auxiliaries and threshold four. Uniform calibration bisects the Bernoulli probability for 80 iterations and sets $\lambda = -16 \log(1-p)$, giving ideal rates of approximately 4.073006 and 4.632550 for targets 0.5 and 0.6. The dynamic program accumulates in FP32; inputs, outputs and the rate buffer follow configured native precision.

The confidence penalty averages tokens within each layer before averaging layers:

$$\mathcal{L} = \mathcal{L}_{task} + \text{sg}\, R_Q + p_Q (R_Q - \text{sg}\, R_Q) \tag{S3}$$

where $R_Q = 0.05\ \text{mean}_l(\text{mean}_t Q_{lt} - q_*)^2$, $q_* = 0.5$ (Global) or 0.6 (Primary).

For a penalty-gradient switch p_Q, use. Full, attachment off, all Q off and attachment only correspond to $(a_Q, p_Q) = (1,1), (0,1), (0,0), (1,0)$. The attachment-only variant sets $p_Q = 0$ while retaining the same forward scalar loss. Ordinary merge, gain and feedback learning remain active; all four settings execute exactly four experts per position.

With fixed incoming adjoint δ_t, the attachment contributes $-2a_Q(\delta_t^\top y_t)(a_5/A)\nabla Q_t$ by main-text Eq. (5), in real arithmetic away from the denominator floor. No inactive expert output is evaluated.

B Fixed-Work Quality and Component Controls

B.1 Training and Evaluation Contract

The CH-Global comparison uses English shard 0022-of-0104 of the locally recorded UltraX-Fine Web corpus, column cleaned_content, and the local allenai/OLMo-7B-hf tokenizer. Rows are traversed in file order, tokenized without added special tokens and terminated with EOS. The 260M quality study uses an equal-token protocol, with 5,000,003,584 training tokens per method, rather than the historical 1.1B fixed-work protocol. Each completed run performs 76,294 updates on four ranks, using local microbatch 16, accumulation two and sequence length 512. Parameters and AdamW moments remain FP32, with BF16 autocast during training.

With exactly 65,536 token exposures per global update (microbatch 16, accumulation two, four ranks), training utilizes BF16 and grouped-BMM expert execution. AdamW uses learning rate

5×10^{-5}, weight decay 0.01, and gradient clipping at one. Learning-rate decay and training cessation strictly follow a fixed token horizon. Training concludes immediately after processing exactly 5,000,003,584 tokens across all methods. Under this shared protocol, CH-Global and Default MoE execute exactly 80,000,057,344 calls, whereas ReMoE's adaptive routing accumulates approximately 80.32B calls (a 0.412% natural computational overshoot).

Each method receives three development candidates at seed 19012; eligible validation NLL selects the recipe before downstream scoring and seeds 19013-19015 evaluate the frozen choices. CH-Global selects attachment scale 2, ReMoE adaptation factor 1.05, and Default MoE scored EMA decay 0.90. ReMoE must end within 4.0 ± 0.1 active experts and satisfy that band in at least 90 of its last 100 steps; all confirmation runs pass. For the Full recipe, scales $\kappa \in \{0.5, 1, 2\}$ span only 0.000647 nat in development NLL (0.0647% in PPL); this is a one-seed, 40-batch sensitivity check. The complete nine-candidate record is supplied electronically.

Full validation uses 999,936 input tokens and 997,983 next-token losses. The short load probe is common across methods and contains 40,960 tokens. Macro accuracy equally weights RACE-High, MMLU, BoolQ, RECORD, ARC-Challenge, and SocialIQA. Seed is the independent endpoint unit; mean $\pm$ sample SD summarizes three seeds. Relative PPL and load-CV improvements average within-seed percentages, avoiding ratios of rounded means.

Table S2: Fixed-work confirmation endpoints. CH-Global uses Full (11), with attachment and penalty gradients enabled. Calls sum four sparse layers. Macro equally weights six tasks. Training concludes exactly at the 5B token horizon for all methods. Results report the exact mean across seeds 19013, 19014, and 19015.

Method	Calls (B)	Mean PPL	Mean Macro (%)
CH-Global (Ours)	80.00	24.527	37.940
ReMoE	80.32	24.677	37.761
Default MoE	80.00	25.000	37.375

B.2 Evidence Supporting the Quality Advantage

Table S2 reproduces the mean paired PPL improvements and the 0.180/0.566-point macro accuracy gains over ReMoE/Default MoE. Mean paired load-CV reductions are 85.23%/96.09%, with all three seeds favorable against both controls. All 54 task cells remain in `evidence/fixed_work_tasks.csv`. The five-task average excluding BoolQ improves by +0.349 points against Default MoE, while performing marginally below ReMoE (-0.060 points); the primary metric retains all six tasks. These baseline comparisons, including main-text Table 1, evaluate Full (11). These baseline comparisons, including main-text Table 1, evaluate the combined Full (11) configuration on the primary reading comprehension suite. The penalty-free Attachment-only (10) configuration from main-text Table 2 is subsequently matched against these external baselines on the secondary frozen task suite (as detailed in Section 5.2) to strictly confirm its standalone superiority.

Table S3: Per-task normalized accuracy (%) at the Llama 260M/5B scale. Best results are bolded. This provides the granular breakdown for the macro accuracy improvements discussed in Section 5.1.

Task	CH-Global	ReMoE	Default MoE	Δ ReMoE	Δ Default
RACE-High	**27.3585**	26.9297	26.8153	+0.4288	+0.5432
MMLU	**23.2374**	23.1591	23.0736	+0.0783	+0.1638
BoolQ	**56.3609**	54.9847	54.7095	+1.3762	+1.6514
ReCORD (EM)	**63.3300**	63.3000	62.6000	+0.0300	+0.7300
ARC-Challenge	19.5392	19.9659	**20.3072**	-0.4267	-0.7680
SocialIQA	37.8199	**38.2293**	36.7451	-0.4094	+1.0748

B.3 Component Controls at the Same Hard Budget

The completed four-arm comparison at seeds 19701, 28463, and 39173 separates the attachment and penalty gradients: Full (11) retains both, 01 retains penalty only, 00 disables both, and 10 retains attachment only. Initialization, inputs, and exact 80B call budgets are rigidly paired within each seed configuration.

Table S4 details the absolute endpoints for every seed and metric. The data exposes a critical optimization dynamic: the presence of the proxy penalty (Arms 01 and 11) systematically inflates the Test PPL (ranging from 26.919 to 26.962). Discarding the penalty entirely (Arm 10) universally corrects this generalization gap across all evaluated seeds, consistently achieving the lowest Test PPL while concurrently driving the Test Load CV to its absolute minimum.

Table S4: Complete four-arm component ablation endpoints at the 260M/5B scale. Best results per seed within each generalization metric are bolded.

Seed	Arm	Test PPL ↓	Macro Acc. (%) ↑	Test CV ↓
19701	00	26.894	45.793	0.03325
19701	01	26.930	45.447	0.03429
19701	10	**26.885**	**45.937**	**0.03000**
19701	11	26.919	45.186	0.03797
28463	00	26.918	44.807	0.03748
28463	01	26.951	**45.515**	0.03611
28463	10	**26.912**	44.751	**0.03467**
28463	11	26.946	44.965	0.04104
39173	00	26.986	45.171	0.03498
39173	01	26.962	44.944	0.03455
39173	10	**26.917**	**45.264**	**0.03326**
39173	11	26.935	44.785	0.03378

To assess AttachOnly against external baselines under a common evaluation protocol, we compare the seed 19701 checkpoints on the original six-task suite using identical frozen scorers and precision settings (FP32 parameters and BF16 autocast). Note that Test PPL is evaluated at sequence length 512 with batch size 16, while the six-task Macro accuracy is evaluated at sequence length 256 with batch size 8. As shown in Table S5, AttachOnly achieves a Test PPL of 26.884716 and a macro accuracy of 45.936649%, compared with 27.083944 and 45.319852% for ReMoE, and 27.452850 and 45.257579% for Default MoE. The macro improvements are 0.617 and 0.679 percentage points, respectively. Relative to All-Q-Off, the attachment reduces Test PPL by 0.009384 and improves macro accuracy by 0.144 percentage points. These descriptive single-seed results provide a matched external-baseline comparison, but do not establish cross-seed robustness or quality-preserving acceleration under the deployment configuration.

Table S5: Single-seed (19701) matched external comparison at the Llama 260M/5B scale. Evaluated on the frozen six-task suite using identical scorers, FP32 parameters, and BF16 autocast. Test PPL uses $L = 512, B = 16$; Macro Accuracy uses $L = 256, B = 8$.

Method	Test PPL ↓	Six-Task Macro (%) ↑
AttachOnly (10)	26.884716	45.936649
All-Q-Off (00)	26.894100	45.792711
Full (11)	26.919269	45.186034
ReMoE	27.083944	45.319852
Default MoE	27.452850	45.257579

C Full-Sequence Execution and Deployment

C.1 Single-A100 Execution and the Equal-Work Comparator

The frozen 4.38B checkpoints (seeds 42/123/777) share BF16, length 512, batch 8, capacity factor 1.25 and a dynamic-Triton executor. Timing includes routing, experts and merge (Table S6). At 120M, throughput improves by 17.16% over Top-4 and 11.22% over same-work Scalar-90; against Scalar-90, dynamic energy falls 19.81% and P50/P90/P99 fall 10.05%/9.97%/10.94%. These compare unrounded means. At 240M, mean paired improvements are 26.03% for throughput and 19.84% for energy.

Table S6: Single-A100 absolute endpoints: mean ± sample SD over three checkpoint seeds. Each checkpoint contributes the median of five rounds (50 warmups and 1,000 timed forwards per round). Energy subtracts idle GPU power. The final row is a separate endpoint evaluation probe.

Metric	Top-4	Scalar-90	CH-Primary
120M-token checkpoints			
P50 (ms)	307.06 ± 6.26	291.31 ± 4.55	262.02 ± 4.13
P90 (ms)	307.79 ± 6.36	292.05 ± 4.45	262.95 ± 4.26
P99 (ms)	312.09 ± 5.37	301.44 ± 5.49	268.46 ± 4.98
Tokens/s	$13,339 \pm 269$	$14,051 \pm 215$	$15,627 \pm 256$
J/token	0.01967 ± 0.00042	0.02156 ± 0.00023	0.01729 ± 0.00029
240M-token checkpoints			
P50 (ms)	322.84 ± 2.42	-	256.18 ± 3.31
P90 (ms)	323.73 ± 2.54	-	256.98 ± 3.41
P99 (ms)	328.47 ± 0.52	-	261.40 ± 6.33
Tokens/s	$12,682 \pm 93$	-	$15,983 \pm 208$
J/token	0.02370 ± 0.00007	-	0.01900 ± 0.00011
Endpoint CV	1.130 ± 0.053	-	0.492 ± 0.037

All three seeds improve all five execution metrics against both controls at 120M. Against Scalar-90, throughput gains range from 8.67% to 15.26% and energy reductions from 19.20% to 20.61% (Table S8). Scalar-90 retains the same branches, balanced-weakest allocation, budget, merge scale and feedback; the contrast evaluates the full confidence-and-routing recipe, including learned assignments and confidence cost. The separate 20,480-position endpoint probe shows 56.39% lower load CV: population CV across sixteen expert counts, averaged over four layers after resetting counters.

C.2 Physical Two-Node and Single-Node Execution

Table S7 uses two distinct hosts and eight verified GPUs; all 36 transport checks and model-equivalence gates pass. Latency is the maximum-rank synchronized full-forward time, with barriers outside timing. Throughput rises 28.66-49.95% and P50 falls 22.72-34.07%; all twelve rounds retain both favorable directions. Rounds repeat one checkpoint pair.

Logical payload falls 10.33-10.64% and receive-route CV falls 63.41-68.32%: modeled remote activation/return/control bytes and rank-level route dispersion, respectively. These accompany the speedup without isolating its causes; no model-level compute-communication overlap was observed.

One eight-A100 node. The same 240M seed-42 pair with packed-metadata execution gives throughput gains of 5.05%/8.60%, P50 reductions of 4.97%/8.02%, and energy reductions of 4.02%/4.64% at batch/rank 2/4. Batch 2 uses five rounds of 1,000 forwards after 50 warmups; batch 4 uses three rounds of 500 after 30. Comparisons are within protocol; two-node energy was not measured.

Memory is the median round maximum of rank-local allocated-tensor peaks, not reserved or whole-device usage. Expert CV is the four-layer mean of population CV on globally summed expert counts: an untimed diagnostic, identical across orders and distinct from receive CV or training load.

Table S7: Two-node execution of the 240M seed-42 pair: four A100s per host, length 512, NCCL Socket. T is Top-4 and C is CH-Primary. Direction denotes the sender-to-receiver host sequence; metric pairs list $T \rightarrow C$. Values are medians of three rounds (10 warmups, 50 forwards each). The lower panel is identical across orders.

Order	B/rank	Tokens/s	P50 (ms)	P99 (ms)	Gain
1	1	13,526 → 18,136	301.44 → 224.95	324.45 → 247.06	34.08%
2	2	13,998 → 20,909	577.87 → 390.71	641.20 → 423.46	49.37%
1	1	15,269 → 19,645	264.97 → 204.77	362.81 → 248.19	28.66%
2	2	17,856 → 26,775	458.64 → 302.39	481.65 → 406.89	49.95%

Table S8: Seed consistency and resource diagnostics for CH-Primary. Upper panel: 120M single-A100 reductions/gains relative to Scalar-90; seeds are the replicate unit. Lower panel: 240M seed-42 Top-4 (T) versus CH-Primary (C), with per-rank batch b.

Seed	Throughput gain (%)	Dynamic energy reduction (%)
42	15.26	20.61
123	8.67	19.63
777	9.85	19.20
Deployment Metrics		
Deployment	Metric ($T \rightarrow C$)	Reduction
One node, $b = 2$	Peak allocated memory: 8480.08 → 8321.80 MiB	1.87%
One node, $b = 4$	Peak allocated memory: 8868.50 → 8552.30 MiB	3.57%
Two nodes, $b = 1$	Expert-load CV: 1.096389 → 0.430555	60.73%
Two nodes, $b = 2$	Expert-load CV: 1.086647 → 0.425878	60.81%

C.3 Batch-Scored Validation and Task Scope

The final policy also lowers validation batch-score PPL from 137.00 to 132.54 at 120M tokens and from 42.985 to 42.056 at 240M. The reductions are 3.26% (ratio of three-seed means) and 2.16% (mean paired seed reduction), respectively. These are batch-coupled supplied-sequence scores, not strict autoregressive likelihoods.

Task scoring truncates context-continuation strings to length 256, sorts by length, left-pads batches of 16, and minimizes length-normalized continuation NLL. Loss excludes context and padding, while all padded positions participate in routing. Main-text Section 5.3 reports the updated task-accuracy gains. To provide full transparency into this representation improvement, Table S9 details the granular performance across all seven evaluated reasoning tasks at the 240M horizon. The breakdown confirms that the algorithmic frugality of CH-Primary actively complements full-sequence optimization, achieving strict improvements in six out of seven tasks without inflating the exact hardware execution budget.

C.4 Single-A100 Shape Sweep Diagnostics

To establish the robust scaling of our offline full-forward implementation across varied workloads, we execute a comprehensive shape sweep on a single A100 (80GB, 400W). We evaluate 24 distinct shape combinations comprising sequence lengths $L \in \{256, 512, 1024, 2048\}$ and batch sizes $B \in \{1, 2, 4, 8, 16, 32\}$. The protocol strictly executes the complete MoE sublayers (FP32 routing, BF16 outputs) and language-model logits, without labels, backward passes, or KV cache overhead. Each of the 24 records aggregates 10 warmups and 25 timed forwards across four order-balanced rounds.

We explicitly resolve the absolute throughput discrepancy between Table S9 (e.g., 64.17% gain, 709.87 ms Top-4 latency) and the checkpoint evaluations in Appendix C.1 (Table S5, 26.03% gain, 322.84 ms latency). Both protocols evaluate the complete model forward pass, including language-model logits. The absolute latency difference arises from distinct execution backends. Table S6 utilizes a highly optimized dynamic-Triton executor. In contrast, the shape sweep in Table S10 uses a

Table S9: Per-task macro accuracy (%) at the 4.38B scale (240M token horizon). CH-Primary demonstrates a net positive gain over the Top-4 baseline across the majority of reasoning tasks, achieving an average improvement of $+0.624$ percentage points.

Task	Top-4	CH-Primary	Δ CH-Primary
RTE	50.0602	**51.8652**	$+1.8051$
SciQ	44.8000	**45.5333**	$+0.7333$
CommonSenseQA	22.9047	**23.7783**	$+0.8736$
SocialIQA	**37.4787**	37.3934	-0.0853
QASC	13.5349	**13.7509**	$+0.2160$
LogiQA	22.3246	**22.9391**	$+0.6144$
RACE-High	24.6903	**24.8999**	$+0.2096$

native PyTorch `grouped_bmm` backend with FP32 expert execution (TF32 disabled) to expose raw architectural scaling. The amplified 64.17% relative gain in Table S9 occurs because `grouped_bmm` structurally pads inputs to the maximum expert load. Since CH-Primary dramatically reduces the maximum expert load (e.g., from 3.913 to 1.969 in our records), it massively eliminates padding overhead in this backend. Together, these protocols prove that our routing topology accelerates both highly fused Triton environments (26.03%) and padding-sensitive native backends (64.17%).

As detailed in Table S10, CH-Primary strictly outperforms the standard Top-4 routing baseline in all 24 evaluated shapes. The pooled throughput gain spans from 9.82% to a peak of 70.37% (at $L = 2048, B = 8$). Furthermore, CH-Primary consistently reduces the peak allocated GPU memory across every configuration, collapsing the maximum memory footprint at the heaviest load ($L = 2048, B = 32$) from 62.901 GiB to 46.828 GiB. These findings definitively prove that the decoupled allocation mechanism scales efficiently across the complete arithmetic-precision grid.

C.5 AUTOREGRESSIVE GENERATION PROFILING

While the primary evaluations in Section 5.3 focus on high-throughput, offline full-sequence processing, we additionally profile the framework's behavior under strict autoregressive text generation. To ensure absolute parity, both the baseline (Top-4) and CH-Primary are evaluated using a highly optimized V2 deployment stack: routing with CUDA Graphs, native DynamicCache, FP32 `grouped_bmm` expert execution (TF32 disabled), and strict omission of EOS tokens to enforce exactly 128 generated tokens per sequence. The hardware is a single 400W A100-SXM4-80GB, evaluating the 240M-token 4.38B dense-equivalent checkpoints. Crucially, to verify that these aggressive system-level optimizations do not alter the underlying arithmetic, we strictly validated the V2 deployment stack against a native reference implementation. Across an exhaustive verification suite (comprising 64 configuration groups and 12,565 archived member files), the optimized execution demonstrated exact cryptographic hash consistency for both the step-by-step full vocabulary logits and the generated tokens. This definitively proves that the measured autoregressive speedups reflect pure systems engineering efficiency, maintaining absolute mathematical fidelity to the unoptimized implementation.

Metrics strictly separate the Time-to-First-Token (TTFT, prefill phase) from the Time-Between-Tokens (TBT, decode phase). As shown in Table S11, the autoregressive performance profile heavily depends on the deployment batch size:

Comprehensive Acceleration at Batch 16 and 32: Under denser workloads ($B = 16$ and $B = 32$), CH-Primary exhibits robust, comprehensive advantages. It significantly reduces TTFT (by 10.69% to 11.71%) and simultaneously improves TBT (by 3.07% to 6.53%). This dual optimization yields up to an 8.13% end-to-end generation throughput increase, proving that our batch-coupled allocation excels in denser concurrent workloads. We explicitly note that due to single-A100 hardware memory constraints, sequence lengths beyond 512 at Batch 32 were omitted.

Prefill-Dominated Acceleration at Batch 8: Under $B = 8$, CH-Primary delivers consistent TTFT reductions (up to 11.84%) but maintains relatively neutral TBT performance improvements (ranging from 0.24% to 0.74%) compared to Top-4. Therefore, while the end-to-end generation throughput

Table S10: Single-A100 full-forward shape sweep (4.38B dense-equivalent scale). CH-Primary universally increases throughput (tok/s), reduces latency (ms), and lowers peak allocated memory (GiB) across all 24 combinations of Sequence Length and Batch Size.

2*Length	2*Batch	Throughput (tok/s) ↑		2*Gain (%) ↑	Latency (ms) ↓		Memory (GiB) ↓	
		Top-4	CH-Pri		Top-4	CH-Pri	Top-4	CH-Pri
256	1	4475.6	4914.9	9.82%	57.20	52.09	12.320	12.258
256	2	5177.3	7339.4	41.76%	98.89	69.76	12.519	12.355
256	4	5561.1	8397.2	51.00%	184.14	121.95	12.918	12.586
256	8	5688.8	9198.0	61.69%	360.01	222.66	13.715	13.051
256	16	5785.4	9514.0	64.45%	707.99	430.52	15.297	14.030
256	32	5838.6	9522.6	63.10%	1403.08	860.27	18.470	16.040
512	1	5159.9	6956.1	34.81%	99.23	73.60	12.519	12.371
512	2	5552.5	8124.7	46.32%	184.42	126.04	12.917	12.597
512	4	5727.6	9144.4	59.66%	357.57	223.96	13.715	13.058
512	8	5770.1	9472.5	64.17%	709.87	432.41	15.302	14.043
512	16	5827.0	9639.7	65.43%	1405.87	849.82	18.468	16.058
512	32	5880.4	9662.8	64.32%	2786.22	1695.58	24.807	20.171
1024	1	5528.5	7862.2	42.21%	185.22	130.24	12.917	12.612
1024	2	5701.9	8949.9	56.96%	359.18	228.83	13.708	13.075
1024	4	5745.2	9611.7	67.30%	712.94	426.15	15.302	14.049
1024	8	5789.3	9819.6	69.62%	1415.03	834.25	18.476	16.035
1024	16	5857.2	9746.6	66.40%	2797.25	1681.00	24.808	20.279
1024	32	5894.4	9664.2	63.96%	5559.18	3390.66	37.479	28.683
2048	1	5651.3	8476.3	49.99%	362.40	241.61	13.710	13.145
2048	2	5700.8	9030.5	58.41%	718.49	453.57	15.302	14.206
2048	4	5747.9	9570.8	66.51%	1425.21	855.93	18.476	16.245
2048	8	5810.7	9899.7	70.37%	2819.63	1655.00	24.834	20.476
2048	16	5848.2	9645.2	64.93%	5603.14	3397.34	37.518	29.287
2048	32	5863.5	9471.3	61.53%	11176.93	6919.46	62.901	46.828

still improves by up to 5.94% for long prompts ($L = 2048$), this gain is fundamentally driven by prefill efficiency rather than token-by-token decode acceleration.

Table S11: Autoregressive generation profiling under the optimized V2 deployment stack (128 tokens generated per sequence). Positive percentages indicate CH-Primary outperforms Top-4. End-to-end metrics include both prefill (TTFT) and decode phases. Missing combinations at $B = 32$ indicate out-of-memory limits on a single 80GB A100.

Batch	Prompt L	TTFT Reduction	TBT Reduction	Decode Tok/s Gain	E2E Tok/s Gain
8	128	8.93%	0.74%	0.75%	1.70%
8	512	11.17%	0.50%	0.50%	3.66%
8	1024	11.54%	0.28%	0.28%	4.75%
8	2048	11.84%	0.24%	0.24%	5.94%
16	128	10.69%	6.00%	6.38%	7.29%
16	512	11.71%	4.36%	4.55%	7.54%
16	1024	11.71%	3.07%	3.17%	7.51%
32	128	11.05%	6.53%	6.99%	8.13%
32	512	11.54%	3.90%	4.06%	7.61%

D ANALYTICAL PROPERTIES AND LIMITATIONS

D.1 LOCAL QUALITY PERTURBATION BOUND

To rigorously understand the representational impact of dropping experts to meet hard execution budgets, we analyze the local quality perturbation. Fixing a given checkpoint, four candidate experts, and their respective outputs, replacing the fourth expert (with weight w_d) and renormalizing according to our merge scale of 4 yields the following bound:

$$||y_4 - y_3|| \leq 4 w_d D_e \tag{S4}$$

where D_e represents the maximum distance between the outputs of the candidate experts. This inequality explicitly demonstrates that safely dropping an expert depends on the dropped weight being small ($w_d \to 0$) and the candidate expert outputs being highly similar ($D_e \to 0$). We recommend that future work supplements this analytical bound with empirical diagnostics of Q versus the actual marginal utility of the fourth expert. However, we strictly note that this localized formula cannot directly explain the empirical macro accuracy gains (e.g., the +0.624 percentage point net improvement across reasoning tasks) observed for CH-Primary over Top-4 at the 240M scale, as the dynamic allocation architecture may implicitly provide beneficial representation regularization during full-sequence optimization.

D.2 TOPOLOGY-AWARE COMMUNICATION COST MODEL

The empirical system speedups observed in distributed settings can be further contextualized using a topology-aware communication cost model. Such an analytical model incorporates NCCL startup overheads, link byte volumes, device compute capacities, and extra routing overheads to establish conditional theoretical acceleration thresholds. However, as explicitly noted in Appendix C.2, we did not observe model-level compute-communication overlap in our physical measurements. Therefore, this topology-aware cost model serves strictly as a predictive framework requiring future verification in heavily optimized environments; the mask closure property and algorithmic allocation alone do not theoretically guarantee perfect communication hiding or zero-overhead distributed execution.

E COMPANION RECORDS AND REPRODUCTION SCOPE

The companion records, including system metrics, source hashes, and a verification script for absolute reproducibility, are temporarily available for double-blind review at `https://anonymous.4open.science/r/CHMOE-25E0`. The full training framework and weights will be open-sourced upon publication. The runtime is PyTorch 2.6.0+cu118/CUDA 11.8 on A100-SXM4-80GB. The 240M run uses eight ranks, local batch one, accumulation four, 14,649 updates, learning rate 1.5×10^{-4} and shards 0021 then 0022. Reruns need original data, tokenizer, weights and runtime; public data/tokenizer revisions are not fully pinned.

F THEORETICAL SCALING PROPERTIES AND PROOFS

This section provides the mathematical proofs for scaling CH-MoE to larger expert pools, strictly focusing on analytical bounds and conditional simplifications while omitting deployment heuristics that fall outside our evaluated scope.

F.1 CONDITIONAL INFERENCE SIMPLIFICATION

For CH-Global at a fixed evaluation state, the auxiliary event Q enters the normalized attachment strictly through $G = \kappa(Q - \text{sg}\, Q)$. Because $G = 0$ during the forward pass, its hard selection depends exclusively on the selection logits, not on Q. Therefore, we prove that if Q has no other forward consumer, an implementation can analytically replace the attachment with its zero value and entirely omit the $O(MN)$ dynamic program during inference, while mathematically guaranteeing identical selected experts and merged outputs.

F.2 Controlled Event Approximation Bound

For excessively large expert pools, evaluating the full-pool dynamic program may introduce arithmetic overhead. We prove a one-sided bound for evaluating the event on a candidate subset S (where $|S| = S_0 \ll M$). Let $p_j = 1 - \exp(-\lambda s_j)$ with independent Bernoulli variables B_j. Defining the subset event $Q_S = \Pr(\sum_{j \in S} B_j \geq K_Q)$ and the omitted probability mass $\tau = \sum_{j \notin S} s_j$, the error is strictly bounded by:

$$0 \leq Q - Q_S \leq 1 - \exp(-\lambda\tau) \leq \lambda\tau. \tag{S5}$$

Proof. The subset event implies the full event. A disagreement requires at least one omitted Bernoulli variable to be active. Independence gives the probability of no omitted activation as $\prod_{j \notin S}(1-p_j) = \exp(-\lambda\tau)$. The final inequality simply applies $1 - e^{-x} \leq x$ for $x \geq 0$.

Furthermore, this yields a sufficient ranking certificate: if the k highest approximate scores form set H, and $\min_{t \in H} \hat{Q}_t > \max_{u \notin H}(\hat{Q}_u + 1 - e^{-\lambda\tau_u})$, then these tokens mathematically form the exact top-k set without requiring full-pool evaluation.

F.3 Logical Work and Tiled Execution Bounds

When transitioning from exact logical calls to hardware-specific block-sparse kernels, backend executors typically round each nonempty expert's token count up to a positive integer tile multiple b. The padded assignment work $\hat{W}$ is analytically bounded against our exact logical budget W:

$$\hat{W} = b \sum_{j : L_j > 0} \lceil L_j / b \rceil, \quad 0 \leq \hat{W} - W \leq A(b-1), \tag{S6}$$

where A is the number of nonempty experts. This proves that mask closure directly restricts maximum padding inflation, bounding worst-case hardware utilization independently of the specific block-sparse kernel behavior.