DELMOE: DUAL-SPARSE ROUTE DELETION WITH RECTIFICATION FOR EFFICIENT BATCHED DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-experts (MoE) models expand model capacity through sparse expert activation without proportionally increasing per-token computation. In batched decoding, different requests can collectively activate many experts, increasing latency without necessarily improving accuracy. Yet skipping experts can reduce accuracy, and surviving gate weights may no longer suit the reduced expert mixture. Existing batch-aware route replacement reduces active experts but largely preserves route counts and redistributes weights without calibrating them for accuracy under the altered routing state. We propose **DELMOE** with two modules for retraining-free batched MoE decoding. (1) **Reuse-aware dual-sparse route deletion:** We build a backbone from high-priority native routes, reuse its experts across requests, and supplement the retained set. Deleting routes outside this set reduces both active experts and token–expert routes. (2) **Routing-aware gate-weight rectification:** We calibrate one rectification coefficient per model offline and keep it fixed during inference to adjust surviving gate weights. To avoid executing deleted zero-weight slots, our custom vLLM CUDA backend processes retained routes as variable-length workloads. Across three MoE models and eight benchmarks, DELMOE achieves higher average accuracy and generally faster decoding than batch-aware route-replacement baselines, with particularly large accuracy gains at high sparsity. Compared with Vanilla, DELMOE achieves up to $2.49\times$ decoding speedup at comparable or higher average accuracy while activating fewer experts.

1 INTRODUCTION

Sparse mixture-of-experts (MoE) models route each token to a small subset of experts and combine their outputs using gate weights. By expanding model capacity while activating only a fraction of the parameters, MoE can improve predictive quality without proportionally increasing per-token computation (Shazeer et al., 2017; Fedus et al., 2022). Although each request generates tokens sequentially, practical serving batches tokens from multiple requests to improve hardware utilization and decoding throughput (Kwon et al., 2023). Different requests can select different experts, so per-token sparsity need not yield a small batch-level active-expert set.

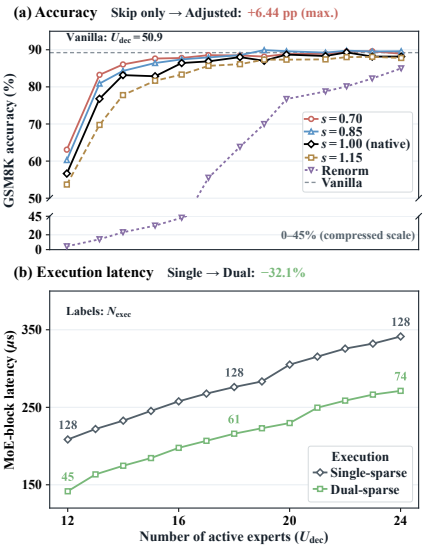

Figure 1: Qwen3: (a) Accuracy under different gate-weight rules; (b) single- vs. dual-sparse latency versus U_{dec}.

However, in batched decoding, additional expert activations increase latency without necessarily improving accuracy. Figure 1(a) reports the accuracy of Qwen3-30B-A3B-Instruct-2507 (Qwen3; Yang et al., 2025) on GSM8K (Cobbe et al., 2021). When lower-ranked experts are skipped while retaining native gate weights (black curve), accuracy largely plateaus at around 20 active experts, approaching Vanilla with roughly 30 fewer. This redundancy incurs both weight-access and execution costs: more active experts (U_{dec}) require accessing more expert weights, while more token–expert routes (N_{dec}) increase execution overhead.

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.

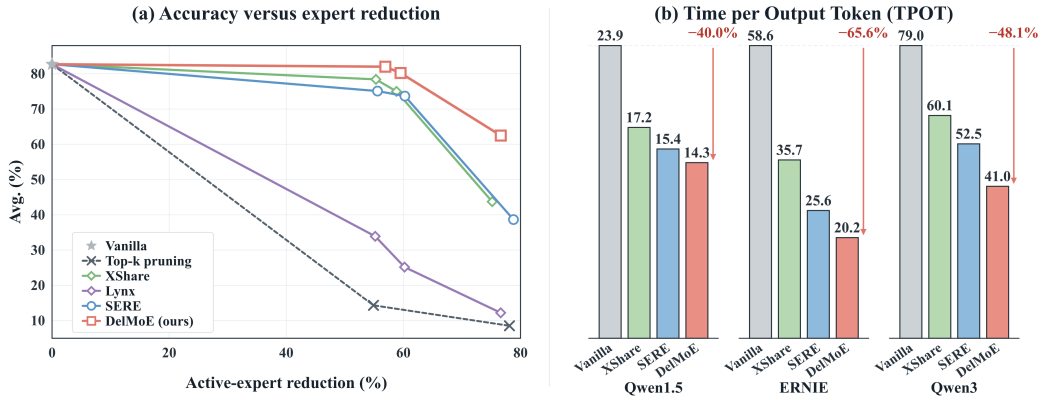

Figure 2: (a) Qwen3 accuracy versus active-expert reduction, $100(1 - U_{\text{dec}}/U_{\text{dec}}^{\text{Vanilla}})\%$. (b) Three-model $k = 1$ Mean TPOT at QPS$= 16$: Bars are normalized to Vanilla, labels show TPOT in milliseconds, and annotations indicate DELMOE reductions.

Figure 1(b) shows that reducing active experts lowers latency, while executing only retained routes further reduces MoE-block latency by up to 32.1% at the same U_{dec}. These observations motivate dual sparsity: jointly reducing the number of active experts and the number of token–expert routes.

Expert skipping can reduce accuracy, and the surviving gate weights may no longer be appropriate. Skipping changes the combined expert output and downstream hidden states, and renormalizing the surviving weights to sum to one can cause substantial additional accuracy loss (purple curve in Figure 1(a)). Counterintuitively, simply attenuating the surviving gate weights improves Qwen3's accuracy on GSM8K by up to 6.44 points over retaining native gate weights at the same expert budget (red curve), with negligible additional computation. Amplifying these weights can instead reduce accuracy (brown curve). At some budgets, attenuation even surpasses Vanilla with far fewer active experts. These observations motivate rectifying the surviving gate weights to account for the routing changes introduced by expert skipping.

Existing batch-aware routing methods only partially address these efficiency and accuracy challenges (Gupta et al., 2024). Their route-replacement strategies redirect tokens from skipped experts to retained ones, reducing the number of active experts while largely preserving token–expert routes and their execution costs (Wu et al., 2026a; Oncescu et al., 2026; Vankov et al., 2026). Moreover, weight transfer or renormalization redistributes native gate mass among retained experts (Figure 3), which may further degrade accuracy.

We propose **DELMOE**, a retraining-free method for fast and accurate batched MoE decoding with two modules. (1) **Reuse-aware dual-sparse route deletion:** Route deletion removes selected token–expert assignments without redirecting them to other experts. We retain each token's high-priority native routes as a backbone, reuse their experts across requests, and supplement the expert set using reuse and mean gate weight. Deleting native routes and gate weights outside this set reduces both U_{dec} and N_{dec}. (2) **Routing-aware gate-weight rectification:** We calibrate one rectification coefficient per model offline and keep it fixed during inference to adjust affected tokens' surviving gate weights. To translate route deletion into execution savings, we implement a custom vLLM CUDA backend that removes the zero-weight slots retained by the fixed-shape execution path before dispatch and processes the remaining routes as variable-length workloads, lowering latency while preserving identical routes and outputs.

Our main contributions are summarized as follows:

1. We propose DELMOE, a retraining-free method for batched MoE decoding that combines dual-sparse route deletion with gate-weight rectification to address activation redundancy and accuracy loss after route deletion.
2. We implement a custom vLLM CUDA backend that realizes dual-sparse execution by skipping zero-weight slots and processing retained routes as variable-length workloads.
3. We evaluate DELMOE on three MoE models and eight benchmarks, showing higher average accuracy at comparable U_{dec} and generally lower latency than route replacement, with up to $2.49\times$ speedup over Vanilla at comparable or higher average accuracy (Table 1; Figures 2 and 5).

2 Related Work

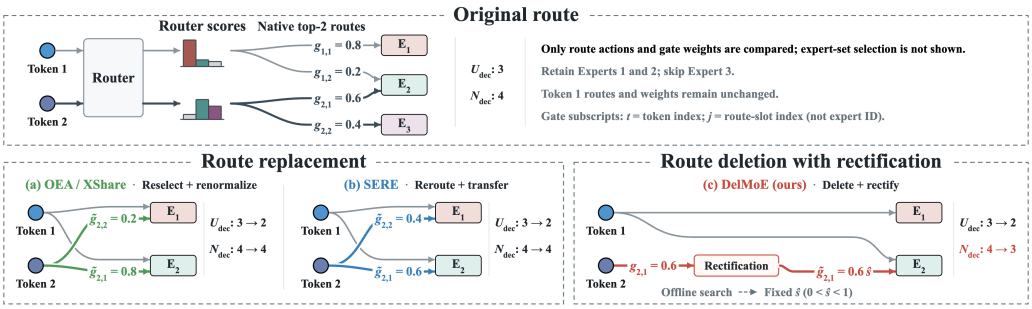

Figure 3: Route replacement versus route deletion with rectification for the same retained experts.

We focus on dynamic expert skipping, rather than static expert compression that alters the deployed model (Xie et al., 2024; Liu et al., 2024; Chen et al., 2025; Zhou et al., 2025; Yoon & Choi, 2026).

Expert skipping and route sparsity. Token-level skipping (Lu et al., 2024; Huang et al., 2025) and training-based route-width adaptation (Huang et al., 2024; Zeng et al., 2024; Wu et al., 2026b) reduce per-token computation, but uncoordinated expert choices can leave many experts active across a batch. We therefore mainly compare with batch-aware methods exploiting reuse (Gupta et al., 2024), alongside direct top-k reduction. OEA/XShare reselect routes by router scores (Oncescu et al., 2026; Vankov et al., 2026); SERE redirects them to similar retained experts (Wu et al., 2026a). At fixed per-token width, replacement reduces U_{dec} but preserves N_{dec} (Figure 3(a–b)).

Expert weighting after skipping. OEA and XShare renormalize surviving weights to sum to one (Oncescu et al., 2026; Vankov et al., 2026), while SERE transfers native weights to retained experts (Wu et al., 2026a). Preserving or redistributing native gate mass does not ensure that surviving weights remain appropriate for the reduced expert mixture. Neither rule explicitly calibrates these weights for accuracy under the altered routing state. Other work separates route width from its normalization reference (Chen & Yao, 2026) without examining batch-aware skipping, or omits renormalization after zero-expert selection through self-distillation (Lv et al., 2026). DelMoE instead uses offline calibration to select one rectification coefficient per model and applies it according to each token's routing state after route deletion.

Efficient MoE execution. Tutel adapts parallelism and pipelining to dynamic expert workloads (Hwang et al., 2023), while MegaBlocks enables dropless block-sparse execution (Gale et al., 2023). Integrating route deletion additionally requires eliminating zero-weight slots retained by our evaluated fixed-shape vLLM path. This requires low-overhead filtering and expert-grouped metadata reconstruction compatible with CUDA Graph replay, motivating our custom CUDA backend.

3 Method: DelMoE

DelMoE combines reuse-aware dual-sparse route deletion with routing-aware gate-weight rectification (Figure 4). Offline calibration selects one rectification coefficient per model, shared across all operating points and held fixed during inference. Both prefill and decoding then delete routes outside the retained expert set and rectify affected tokens' surviving gate weights. Our custom vLLM CUDA backend executes retained routes as variable-length, expert-grouped workloads.

3.1 Reuse-Aware Dual-Sparse Route Deletion

Let $\mathcal{T}$ denote the non-padding tokens in a batched MoE call. For each $t \in \mathcal{T}$, the native router returns K ordered experts $e_{t,j}$ and positive gate weights $g_{t,j}$, with $j = 1$ highest-ranked. Their native token–expert routes form $\mathcal{R}$. For expert feed-forward network (FFN) F_e and layer input h_t, the routed-branch output is $y_t = \sum_{j=1}^{K} g_{t,j} F_{e_{t,j}}(h_t)$; shared experts are unchanged. We denote the number of distinct active experts by U_{dec} and the number of token–expert routes by N_{dec}. Reported values average these quantities over layers and decoding steps.

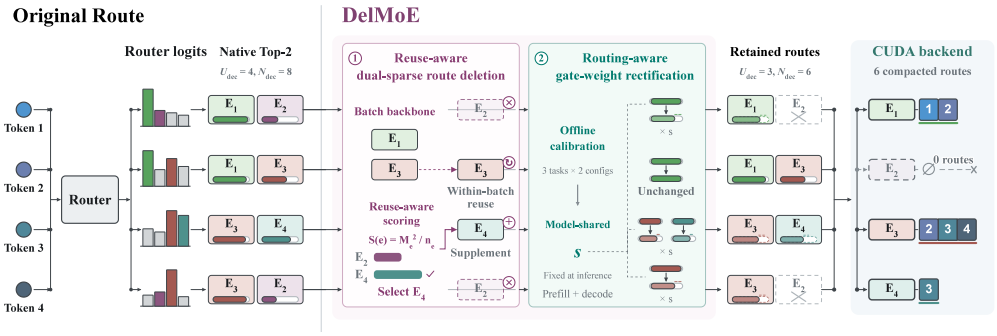

Figure 4: DELMOE on four tokens with native width $K = 2$, backbone width $k = 1$, and supplemental expert budget $b = 1$. Reuse-aware selection deletes out-of-set routes. Offline calibration selects a model-shared rectification coefficient for gate-weight rectification in both prefill and decoding. CUDA executes six retained token–expert routes as variable-length workloads.

DELMOE retains each token's first k native routes as a backbone, where $1 \le k \le K$. The union of their expert IDs across the batch forms P, the initial deletion boundary. Any later-rank native route into P is also retained through within-batch reuse, without consuming the nonnegative integer supplemental expert budget b. Thus, even $b = 0$ can retain more than k routes per token, while requiring no experts beyond the backbone.

Experts appearing in later-rank native routes but absent from P form the candidate set C, from which at most b supplemental experts are selected; invalid IDs are ignored.

For each $e \in C$, let n_e count its later-rank native routes and M_e be their gate-weight sum. We rank candidates by

$$S(e) = \frac{M_e^2}{n_e}. \tag{1}$$

The score increases with reuse count or mean gate weight when the other is fixed. It ranks candidates within the batch without measuring task-level importance.

The $\min(b, |C|)$ highest-scoring candidates form the supplemental set A, with ties favoring the smaller expert ID. The final deletion boundary is $H = P \cup A$. DELMOE retains exactly the native routes into H and deletes all routes and gate weights outside it. Supplementation adds expert IDs to the retained set without constructing replacement routes, so the retained route set satisfies $\mathcal{R}' \subseteq \mathcal{R}$. Every token retains its first k routes, while its total retained route count can vary with within-batch reuse and supplementation.

After deletion, $U_{\text{dec}} = |H| = |P| + \min(b, |C|)$ and $N_{\text{dec}} = |\mathcal{R}'|$. Within-batch reuse determines $|P|$, while b caps expert-level additions for finer control than increasing every token's backbone width k. Deleting out-of-set experts also removes their native routes, jointly reducing U_{dec} and N_{dec}. Appendix A.1 details rectification coefficient selection and single-call routing (Algorithm A1).

3.2 ROUTING-AWARE GATE-WEIGHT RECTIFICATION

Deleting routes changes the expert mixture, so surviving gate weights may need rectification. We control their attenuation with $s \in (0, 1]$, without updating model parameters. At $s = 1$, surviving weights stay native and deleted weights remain zero: the *native* rule in Figure 1(a).

We select one rectification coefficient per model through offline calibration on three datasets covering mathematics, general reasoning, and code generation, using calibration examples separate from final evaluation. For each candidate coefficient, we evaluate two configurations, $(k, b) = (1, 0)$ and $(2, 0)$, and maximize the equally weighted mean of the six accuracy scores, breaking ties in favor of the larger coefficient. The selected coefficient is then reused across all operating points of that model and remains fixed throughout prefill and decoding. Appendix B details the calibration datasets and selection protocol.

Before supplementation, we set $d_t = 1$ if token t has any later-rank native expert outside P, and $d_t = 0$ otherwise. Given the retained expert set H and selected rectification coefficient $0 < s \leq 1$, the final weights are

$$\widetilde{g}_{t,j} = \begin{cases} 0, & \text{if } e_{t,j} \notin H, \\ s g_{t,j}, & \text{if } e_{t,j} \in H \text{ and } d_t = 1, \\ g_{t,j}, & \text{if } e_{t,j} \in H \text{ and } d_t = 0. \end{cases} \tag{2}$$

This rule preserves the relative weights of each affected token's surviving experts without additional renormalization; native router normalization and shared experts remain unchanged. For a given input and (k, b), changing s leaves the routes within that MoE call unchanged, although subsequent autoregressive trajectories may diverge. Section 4.4.3 compares renormalization, native gate weights after route deletion ($s = 1$), and rectification with the offline-calibrated model-shared rectification coefficient using mean accuracy across eight evaluation tasks.

3.3 Expert-Grouped Variable-Length CUDA Execution

To convert logical route deletion into less physical work, DelMoE's custom vLLM (Kwon et al., 2023) CUDA backend omits zero-weight slots—token–expert positions in the fixed-width routing representation—before dispatch. In Figure 4 (right), this removes two of eight logical slots and groups the six retained token–expert routes by expert. The backend forms contiguous per-expert workloads, pads each only to the required row-alignment granularity, and scatters weighted outputs back to tokens. These changes affect only physical organization and workload length, preserving the routed output $\widetilde{y}_t = \sum_{j=1}^{K} \widetilde{g}_{t,j} F_{e_{t,j}}(h_t)$ with the weights in Equation 2. Before alignment, each of the N_{dec} retained routes becomes one assignment. The backend may therefore execute more aligned rows than N_{dec}, depending on the number of routes per expert and the alignment granularity. Section 4.3 measures end-to-end effects; Appendix A.2 details CUDA metadata, alignment, buffers, and CUDA Graph behavior.

4 Experiments

4.1 Experimental Setup

Models and baselines. Experiments cover three representative MoE models: Qwen1.5-MoE-A2.7B-Chat (Qwen Team, 2024), ERNIE-4.5-21B-A3B-PT (Baidu-ERNIE-Team, 2025), and Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025). Baselines include Vanilla (native top-K routing), Direct (top-k reduction), Lynx (Gupta et al., 2024), SERE (Wu et al., 2026a), OEA (Oncescu et al., 2026), and XShare (Vankov et al., 2026). We adjust Lynx's α and SERE's ρ to control the measured number of active experts, U_{dec}, and compare operating points with similar U_{dec}. We use official code or faithfully reproduce the original methods, with identical checkpoints and evaluation settings within each model for fair comparison. For Qwen1.5, OEA-NR and XShare-NR preserve native gate weights for compatibility without changing expert selection (Appendix C.1).

Benchmarks and metrics. We evaluate accuracy on eight OpenCompass benchmarks (OpenCompass Contributors, 2023) in three categories: general knowledge and reasoning (CMMLU (Li et al., 2024), BoolQ (Clark et al., 2019), and BBH (Suzgun et al., 2023)), mathematics (MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and Math401 (Yuan et al., 2023)), and code generation (HumanEval (Chen et al., 2021), abbreviated HEval, and MBPP (Austin et al., 2021)). CMMLU and BoolQ use chain-of-thought prompting. For serving evaluation, we measure mean time per output token (TPOT) in vLLM (Kwon et al., 2023) at varying request rates, expressed in queries per second (QPS), with fixed 128/32-token input/output lengths.

Protocols and hardware. We measure accuracy, U_{dec}, and N_{dec} on one NVIDIA A100 GPU and latency on one NVIDIA A800 GPU. Accuracy and latency comparisons between expert-skipping methods use operating points matched by measured U_{dec}. For DelMoE, the rectification coefficient $\widehat{s}$ is selected offline and held fixed during inference. Appendix B provides detailed experimental settings.

4.2 ACCURACY AT MATCHED NUMBERS OF ACTIVE EXPERTS

Table 1: OpenCompass accuracy under nearest-neighbor U_{dec} matching at batch size 16. Bold marks the best available point estimate per column within each model's $k = 1$, $k = 2$, and Sat./Near-sat. groups, including ties.

Group	Method (configuration)	$U_{\text{dec}}\downarrow$	$N_{\text{dec}}\downarrow$	General			Math			Code		Avg.
				CMMLU	BOOLQ	BBH	MATH	GSM8K	MATH401	HEval	MBPP	(Acc. ↑)
Qwen1.5-MoE-A2.7B-Chat (native $K = 4$)												
–	Vanilla	38.25	64.00	69.14	80.12	35.09	13.92	51.25	59.85	48.17	29.40	48.37
$k = 1$	Direct (k=1)	12.98	**16.00**	48.84	38.47	35.38	4.84	26.69	47.38	12.80	14.00	28.55
	Lynx	13.12	64.00	60.26	75.05	26.25	1.10	2.58	24.19	0.00	3.40	24.10
	SERE (k=1,ρ=0)	13.16	64.00	46.93	78.41	30.71	4.54	22.29	46.63	3.66	12.40	30.70
	OEA-NR (k=1)	13.05	63.87	67.10	**79.33**	35.27	**9.70**	**41.62**	**57.11**	32.32	27.00	43.68
	DELMOE (k=1,b=0,s=0.85)	**12.97**	26.61	**67.65**	78.17	**36.69**	8.90	40.18	55.11	**35.98**	**27.60**	**43.79**
$k = 2$	Direct (k=2)	22.92	**32.00**	66.28	73.85	34.42	12.18	44.66	55.86	42.68	27.60	44.69
	Lynx	23.22	64.00	65.70	79.33	32.39	3.82	13.04	46.88	6.71	18.00	33.23
	SERE (k=2,ρ=0)	23.24	64.00	67.34	**80.37**	33.82	12.48	47.99	54.11	38.41	26.80	45.17
	OEA-NR (k=2)	22.99	63.96	**68.83**	79.82	35.24	13.82	49.81	58.60	43.29	29.00	47.30
	DELMOE (k=2,b=0,s=0.85)	**22.91**	43.81	68.65	79.51	**35.87**	**13.88**	**51.10**	**60.35**	**45.12**	**32.20**	**48.34**
Sat.	Lynx	18.13	64.00	63.49	77.83	30.08	1.56	4.85	34.66	0.00	10.80	27.91
	SERE (k=1,ρ=0.285)	**17.94**	64.00	64.60	78.04	33.09	5.72	31.54	51.87	15.24	15.60	36.96
	XShare-NR (k=1,m=5)	17.98	64.00	69.18	**79.79**	36.23	10.78	41.62	56.61	31.71	**31.40**	44.67
	DELMOE (k=1,b=5,s=0.85)	17.95	**36.99**	**69.71**	79.63	**36.28**	**12.60**	**50.87**	**60.10**	**54.27**	29.80	**49.16**
ERNIE-4.5-21B-A3B-PT (native $K = 6$)												
–	Vanilla	41.80	96.00	80.99	86.88	80.09	70.40	90.60	78.30	79.88	69.60	79.59
$k = 1$	Direct (k=1)	12.84	**16.00**	19.77	60.21	25.75	21.40	35.41	38.90	8.54	11.20	27.65
	Lynx	12.61	96.00	78.72	85.14	60.79	39.18	50.95	63.34	18.90	35.80	54.10
	SERE (k=1,ρ=0)	12.34	96.00	79.60	85.14	69.49	56.80	81.50	66.33	71.34	59.20	71.18
	OEA (k=1)	12.41	94.95	77.99	85.54	73.23	61.54	86.20	74.81	76.22	49.60	73.14
	DELMOE (k=1,b=0,s=0.95)	**12.33**	38.91	**81.14**	**86.24**	**74.71**	**67.90**	**89.08**	**78.55**	**79.27**	**68.60**	**78.19**
$k = 2$	Direct (k=2)	21.53	**32.00**	70.87	82.35	70.37	66.22	87.41	75.31	77.44	64.60	74.32
	Lynx	21.56	96.00	**81.09**	**86.97**	73.70	67.36	85.37	76.81	72.56	68.60	76.56
	SERE (k=2,ρ=0)	**21.39**	96.00	80.82	86.09	78.79	70.26	**90.67**	78.30	**80.49**	69.40	79.35
	OEA (k=2)	21.44	95.45	80.84	86.51	**80.31**	66.46	90.30	77.31	79.27	66.80	78.48
	DELMOE (k=2,b=0,s=0.95)	21.47	61.43	81.04	86.27	78.21	**71.42**	90.30	**79.55**	78.66	**69.60**	**79.38**
Sat.	Lynx	14.62	96.00	79.40	85.66	63.78	50.78	61.71	67.58	40.24	50.40	62.44
	SERE (k=1,ρ=0.22)	14.72	96.00	80.07	85.69	70.05	58.36	84.08	69.83	74.39	58.80	72.66
	XShare (k=1,m=2)	14.47	95.55	80.26	**86.82**	**79.51**	63.44	89.54	76.81	75.00	63.40	76.85
	DELMOE (k=1,b=2,s=0.95)	**14.42**	**45.92**	**81.42**	86.76	77.77	**70.58**	**90.60**	**81.55**	**79.88**	**70.20**	**79.84**
Qwen3-30B-A3B-Instruct-2507 (native $K = 8$)												
–	Vanilla	50.90	128.00	84.62	90.00	77.11	72.46	89.23	80.80	89.02	78.40	82.71
$k = 1$	Direct (k=1)	11.14	**16.00**	0.00	61.53	4.64	0.08	1.29	0.75	0.00	0.00	8.54
	Lynx	11.90	128.00	4.46	29.79	21.40	0.18	1.67	36.66	1.83	1.80	12.22
	SERE (k=1,ρ=0)	**10.79**	128.00	51.41	73.98	40.09	28.72	39.35	42.89	17.07	15.80	38.66
	OEA (k=1)	12.64	127.83	65.30	84.95	44.80	28.46	34.04	38.65	38.41	15.20	43.73
	DELMOE (k=1,b=0,s=0.95)	11.89	45.10	**76.77**	**89.36**	**66.21**	**51.82**	**58.68**	**60.85**	**47.56**	**48.60**	**62.48**
$k = 2$	Direct (k=2)	22.96	**32.00**	10.18	59.27	10.14	3.16	5.76	23.19	2.44	0.60	14.34
	Lynx	20.26	128.00	32.30	62.05	34.01	3.14	7.81	46.38	7.93	7.80	25.18
	SERE (k=2,ρ=0)	**20.23**	128.00	78.37	89.02	67.41	64.96	79.23	74.31	81.10	55.00	73.68
	OEA (k=2)	20.96	128.00	80.70	90.09	69.07	64.26	77.63	74.56	**85.98**	57.80	75.01
	DELMOE (k=2,b=0,s=0.95)	20.61	71.82	**83.56**	**90.12**	**74.09**	**71.92**	**87.11**	**81.05**	85.37	**68.60**	**80.23**
Near-sat.	Lynx	22.81	128.00	51.21	72.57	41.14	10.00	21.00	49.38	12.80	13.20	33.91
	SERE (k=2,ρ=0.325)	22.61	128.00	79.87	89.91	67.89	66.08	78.32	80.80	82.93	55.00	75.10
	XShare (k=2,m=2)	22.75	128.00	83.58	**90.21**	72.13	68.12	83.70	75.56	83.54	70.40	78.41
	DELMOE (k=1,b=10,s=0.95)	21.93	85.37	**84.25**	90.03	**75.69**	**72.84**	**89.39**	**83.04**	87.20	**73.60**	**82.01**
	DELMOE (k=2,b=1,s=0.95)	**21.40**	**78.06**	83.92	89.91	74.81	72.66	87.11	82.54	**87.80**	71.60	81.29
	DELMOE (k=2,b=2,s=0.95)	22.58	81.73	83.94	90.03	75.21	72.26	88.32	81.80	85.98	72.60	81.27

Across the three models, DELMOE provides operating points that retain **over 99%** of Vanilla's eight-task average accuracy while reducing active experts by **53.1%–65.5%** and token–expert routes by **33.3%–52.2%**. Direct top-k reduction also reduces both counts, but incurs much greater accuracy loss at comparable U_{dec}.

DELMOE achieves higher eight-task average accuracy than the strongest available baseline in **all nine U_{dec}-matched groups**, with gains of up to **18.75 percentage points**. Its advantage is particularly pronounced under aggressive expert skipping: on Qwen3 at $k = 1$, DELMOE achieves 62.48% average accuracy, compared with 43.73% for OEA, 38.66% for SERE, and 12.22% for Lynx. As supplementary cross-protocol context, its averages also exceed SERE's four published $\rho = 0$ results (Wu et al., 2026a), summarized in Appendix C.1. The matched comparisons also reveal a routing distinction: SERE, OEA, and XShare reduce U_{dec} but yield almost no reduction

in N_{dec}. Route replacement can introduce additional error when selected experts do not reproduce the removed experts' outputs, while weight transfer or renormalization can amplify their weighted contributions. Controlled ablations in Sections 4.4.2 and 4.4.3 support the accuracy benefits of DELMOE's route deletion and moderate attenuation of affected tokens' surviving gate weights.

Fine-grained supplementation provides accuracy–sparsity trade-offs beyond changing k alone. For Qwen1.5 and ERNIE, the Sat. configurations supplement a top-1 backbone and exceed the $k = 2, b = 0$ accuracy with **21.7% and 32.8%** fewer active experts, respectively, and fewer routes. For Qwen3, we report several neighboring operating points in the empirical near-saturation region to characterize the local accuracy–sparsity trade-off, rather than identifying a unique saturation configuration. Section 4.4.1 analyzes expert selection and the choice of Sat. operating points.

4.3 SERVING ACCELERATION AND SYSTEMS ANALYSIS

DELMOE achieves the lowest measured Mean TPOT among the compared expert-skipping methods in **23/27** settings, using dual-sparse route deletion and its custom CUDA execution backend. As shown in Figure 5, DELMOE substantially reduces decoding latency relative to Vanilla across all evaluated QPS values. Across Qwen1.5 and Qwen3, DELMOE achieves speedups of up to **1.96**×. For ERNIE at QPS= 16, the accuracy-preserving Sat. point achieves a **2.49**× speedup, while aggressive $k = 1$ reaches **2.91**× with only a 1.40-point loss in eight-task *Avg.* relative to Vanilla. In the four Qwen1.5 exceptions, DELMOE is 1.5%–7.1% slower than SERE but has a 12.20–13.09-point advantage in eight-task *Avg.* at the matched operating points (Table 1). Appendix C.2 reports complete TPOT and time-to-first-token (TTFT) results, achieved request throughput, and additional diagnostics.

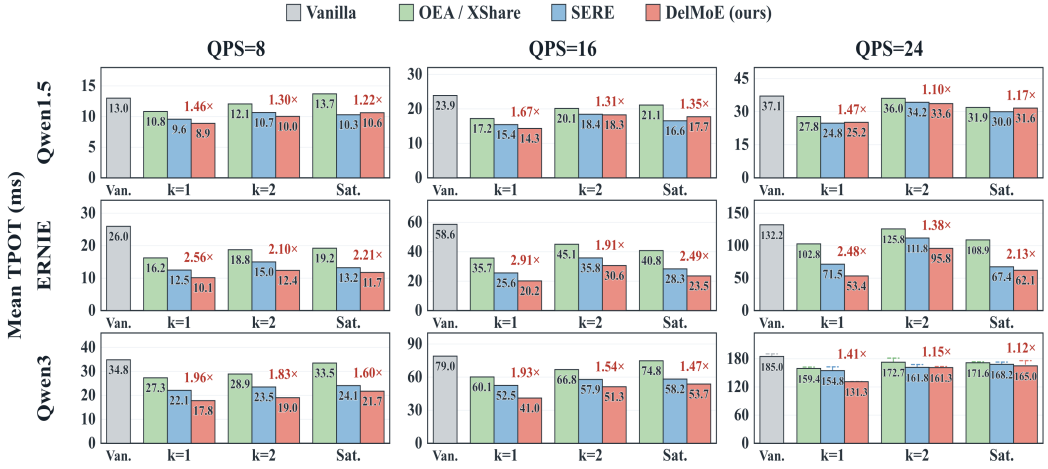

Figure 5: End-to-end Mean TPOT. Speedups are relative to Vanilla for the same model and QPS; dashed Qwen3/QPS= 24 ranges show min–max over three runs.

4.4 MECHANISM ABLATIONS

4.4.1 FINE-GRAINED BATCH-AWARE EXPERT SELECTION

Increasing the backbone width k adds a route to every token, which can activate more experts than needed to recover accuracy. DELMOE instead supplements the batch-level expert set through the budget b, providing finer control over the accuracy–latency trade-off. Figure 6 examines this trade-off using per-point rectification coefficient selection, with the search protocol detailed in Appendix C.3. On Qwen1.5 and ERNIE, supplementing a top-1 backbone achieves higher accuracy with fewer active experts than an unsupplemented top-2 backbone. On Qwen3, $(2, 2)$ achieves comparable accuracy to $(1, 11)$ (82.32% vs. 82.28%) with lower Mean TPOT (60.25 vs. 61.50 ms). The curves show diminishing accuracy gains as more experts are added, motivating $(1, 5)$, $(1, 2)$, and $(2, 2)$ as representative saturation configurations. These results demonstrate the value of allocating a small supplemental expert budget rather than uniformly increasing every token's backbone width.

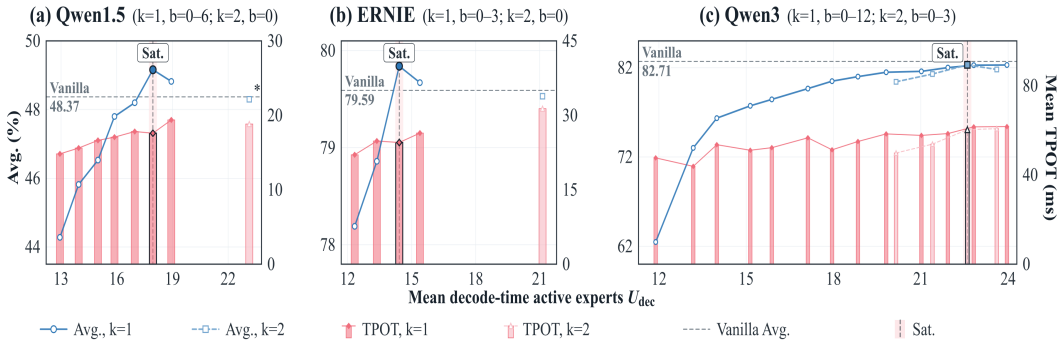

Figure 6: Budget exploration with per-point rectification coefficient selection: eight-task *Avg.* and TPOT (QPS= 16); *$s = 0.80$ system references.

At $s = 1$, M^2/n achieves the highest mean eight-task accuracy (46.16) across five Qwen1.5 budgets ($k = 1, b = 1$–5), while reverse selection ranks last (Table 2). At Qwen1.5's Sat. point $(1, 5)$, applying its model-shared rectification coefficient $s = 0.85$ gives M^2/n the highest accuracy among the tested selectors (49.16) and the largest improvement over its native-weight counterpart (2.13 points). Its strong average performance across Qwen1.5 budgets, leading accuracy at that model's Sat. point with its shared rectification coefficient, and competitive results on the other models support our choice of M^2/n. Although $\max(g)$ performs best at the ERNIE and Qwen3 Sat. points in the $s = 1$ controls, M^2/n stays within one percentage point of the best selector across all tested settings (Appendix C.3).

Table 2: Selector ablation: eight-task *Avg.* (batch size 16; all seven rules in Appendix C.3).

Model	Qwen1.5						ERNIE	Qwen3
(k, b, s)	$(1, 1, 1.00)$	$(1, 2, 1.00)$	$(1, 3, 1.00)$	$(1, 4, 1.00)$	$(1, 5, 1.00)$	$(1, 5, 0.85)$	$(1, 2, 1.00)$	$(2, 2, 1.00)$
Fixed random	44.07	43.98	44.51	44.27	44.28	45.89	77.57	80.43
Reverse M^2/n	43.21	43.43	43.49	43.35	43.13	44.61	77.85	80.49
$\max(g)$	**44.33**	45.16	45.45	**47.46**	**47.32**	47.85	**79.59**	**81.95**
$\boldsymbol{M^2/n}$ **(ours)**	44.16	**45.80**	**46.97**	46.86	47.03	**49.16**	78.68	81.40

4.4.2 Route Deletion vs. Route Replacement

Deletion improves mean eight-task accuracy by 0.87 and 8.60 percentage points over OEA/XShare-style Expert share (Oncescu et al., 2026; Vankov et al., 2026) and SERE-style Expert reroute ($\rho = 0$) (Wu et al., 2026a), respectively (Appendix C.4). Table 3 uses the batch union of native top-k experts ($k = 1, 2$; $b = 0$), matching SERE's primary-set construction without using DELMOE's M^2/n supplementation score. All actions preserve native source-route weights ($s = 1$) without additional renormalization. It outperforms them in four and all six settings, respectively; Expert share leads only on ERNIE. At higher expert sparsity, route deletion provides larger accuracy gains over both replacement actions, on average across models. These controls suggest replacement can introduce mismatched weighted contributions that deletion avoids. Together with the serving results (Figure 5), these findings show that route deletion can achieve lower latency at comparable or higher accuracy by jointly reducing active experts and token–expert routes.

Table 3: Route-action ablation: eight-task *Avg.* at fixed expert sets ($s = 1$, batch size 16).

Action	Qwen1.5		ERNIE		Qwen3		Mean
(k, b)	$(1, 0)$	$(2, 0)$	$(1, 0)$	$(2, 0)$	$(1, 0)$	$(2, 0)$	
Expert share (OEA/XShare-style)	40.94	46.57	**77.94**	**79.93**	59.90	79.83	64.19
Expert reroute (SERE-style)	30.70	45.17	71.18	79.35	38.66	73.68	56.46
Route deletion (ours)	**43.27**	**47.57**	77.43	79.70	**62.21**	**80.13**	**65.05**

4.4.3 Gate-Weight Rectification after Route Deletion

Model-shared rectification raises the nine-point mean accuracy from **66.38% to 66.96%**, improving seven of the nine operating points without separate calibration for each point. Table 4 compares

Renorm, native weights ($s = 1$), and model-shared rectification using eight-task mean accuracy at nine selected operating points in Table 1. The shared rectification coefficients are 0.85, 0.95, and 0.95 for Qwen1.5, ERNIE, and Qwen3, respectively, selected once per model by the offline calibration in Section 3.2 and held fixed during inference. Native weights outperform Renorm at all nine points. Accuracy decreases by 0.32 points at ERNIE $(2, 0)$ and 0.13 points at Qwen3 $(2, 2)$, so the overall improvement does not imply a gain at every point.

Table 4: Gate-weight rules for DELMOE: eight-task mean accuracy at nine selected Table 1 points (BS$=16$). Mean averages nine points; bold: column maxima, including ties.

Rule	Qwen1.5			ERNIE			Qwen3			Mean
(k, b)	$(1, 0)$	$(2, 0)$	$(1, 5)$	$(1, 0)$	$(2, 0)$	$(1, 2)$	$(1, 0)$	$(2, 0)$	$(2, 2)$	
Renorm	25.23	42.60	38.14	66.93	77.14	71.98	21.54	64.79	74.14	53.61
native ($s = 1$)	43.27	47.57	47.03	77.43	**79.70**	78.68	62.21	80.13	**81.40**	66.38
Model-shared	**43.79**	**48.34**	**49.16**	**78.19**	79.38	**79.84**	**62.48**	**80.23**	81.27	**66.96**

4.4.4 VARIABLE-LENGTH VS. FIXED-SHAPE EXECUTION

Our variable-length CUDA backend achieves up to a **1.30**× end-to-end speedup over Fixed-shape across the nine QPS$=16$ pairs (Figure 7(b)). We compare vLLM's fixed-shape MoE execution (Fixed-shape) with our variable-length CUDA backend (Compacted), using identical DELMOE routes, gate weights, and outputs. Although route deletion reduces N_{dec}, Fixed-shape retains zero-weight slots for deleted routes. Compacted omits these slots before dispatch, converting the reduction in N_{dec} into less physical work. In the Qwen3 MoE block ($k = 1, b = 0$, BS$=16$, $N_{\text{dec}} = 45$), this reduces execution slots before alignment from 128 to 45 and lowers MoE-block latency by **31.9%** (Figure 7(a)). Most of this reduction comes from post-gate routing and metadata processing, whose time falls by **73.3%**.

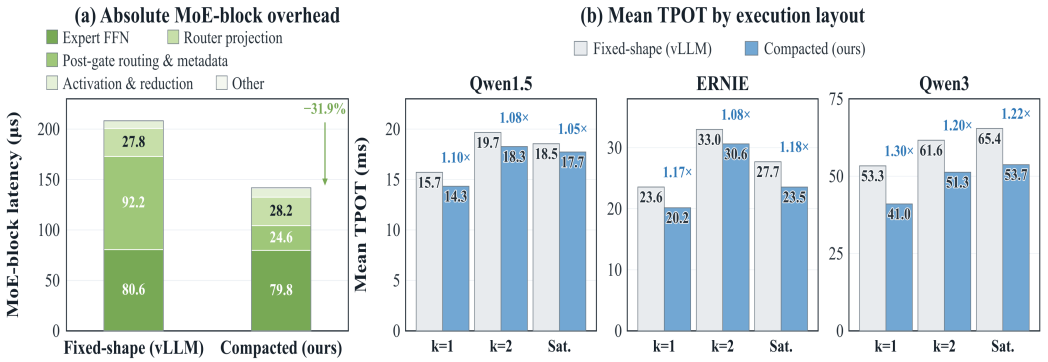

Figure 7: Compacted versus Fixed-shape execution under identical DELMOE routes and outputs: (a) Qwen3 MoE-block latency breakdown; (b) Mean TPOT across nine QPS$=16$ pairs.

5 CONCLUSION

In this work, we investigate two bottlenecks in batched MoE decoding for serving: redundant expert activations that increase latency without necessarily improving accuracy, and accuracy loss when surviving gate weights become mismatched to the routing state after expert skipping. DELMOE addresses these limitations without retraining through reuse-aware dual-sparse route deletion and routing-aware gate-weight rectification. Its custom CUDA backend converts reductions in expert and route counts into execution savings. Across three MoE models and eight benchmarks, DELMOE achieves up to $2.49\times$ speedup over Vanilla at comparable or higher average accuracy. We offer a new perspective on batched MoE decoding: jointly reducing active experts and token–expert routes, while rectifying surviving gate weights according to the routing state after expert skipping, can yield higher accuracy and lower latency.

AI USE STATEMENT

We used large language model (LLM) tools to help improve the manuscript's grammar, clarity, and readability, assist with literature searches, and support the implementation and debugging of experimental code and scripts under author-specified designs and protocols. The authors determined the research ideas, experimental designs, and final analysis and interpretation of the results. The authors have thoroughly reviewed the paper and take full responsibility for its content, the experimental implementation, and the reported results.

REPRODUCIBILITY STATEMENT

The full implementation of DELMOE, including its efficient CUDA kernels, is provided in the supplementary material and will be publicly released upon publication. Section 3 and Appendix A describe the method, pseudocode, and implementation details. Section 4.1 and Appendix B document the model checkpoints, benchmarks, generation settings, hardware and software environments, and measurement protocols. Appendix B also specifies the construction of the three calibration datasets and the procedure for selecting one fixed rectification coefficient per model. Appendix C provides detailed experimental results and ablations, including coefficient search grids and per-task scores. The implementation, evaluation scripts, and calibration inputs are also available at DELMOE.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Baidu-ERNIE-Team. ERNIE 4.5 technical report. Technical report, Baidu, 2025. URL https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf.

I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse MoE via hierarchical clustering. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8594–8620. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25aq.html.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Xing Chen and Hengshuai Yao. Training-free halving of activated experts in fine-grained mixture-of-experts models. *arXiv preprint arXiv:2609.04575*, 2026. URL https://arxiv.org/abs/2609.04575.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

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. URL https://jmlr.org/papers/v23/21-0998.html.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. MegaBlocks: Efficient sparse training with mixture-of-experts. In *Proceedings of Machine Learning and Systems*, volume 5, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/5a54f79333768effe7e8927bcccffe40-Abstract-mlsys2023.html.

Vima Gupta, Jae Hyung Ju, Kartik Sinha, Ada Gavrilovska, and Anand Padmanabha Iyer. Lynx: Enabling efficient MoE inference through dynamic batch-aware expert selection. *arXiv preprint arXiv:2411.08982*, 2024. URL https://arxiv.org/abs/2411.08982.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.696. URL https://aclanthology.org/2024.acl-long.696/.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts LLMs gains more. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hheFYjOsWO.

Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, HoYuen Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong. Tutel: Adaptive mixture-of-experts at scale. In *Proceedings of Machine Learning and Systems*, volume 5, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/5616d34cf8ff73942cfd5aa922842556-Abstract-mlsys2023.html.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://dl.acm.org/doi/10.1145/3600006.3613165.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11260–11285. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.671. URL https://aclanthology.org/2024.findings-acl.671/.

Enshu Liu, Junyi Zhu, Zinan Lin, Xuefei Ning, Matthew B. Blaschko, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs. *arXiv preprint arXiv:2407.00945*, 2024. URL https://arxiv.org/abs/2407.00945.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6159–6172. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.334. URL https://aclanthology.org/2024.acl-long.334/.

Xingtai Lv, Li Sheng, Kaiyan Zhang, Yichen You, Siyan Gao, Xueheng Luo, Yuxin Zuo, Yuchen Fan, Junlin Yang, Ganqu Cui, Bingning Wang, Fan Yang, Youbang Sun, Ning Ding, and Bowen Zhou. Post-trained MoE can skip half experts via self-distillation. *arXiv preprint arXiv:2605.18643*, 2026. URL https://arxiv.org/abs/2605.18643.

Costin-Andrei Oncescu, Qingyang Wu, Wai Tong Chung, Tsai-chuan Wu, Bryan Gopal, Junxiong Wang, Tri Dao, and Ben Athiwaratkun. Opportunistic expert activation: Batch-aware expert routing for faster decode without retraining. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://openreview.net/pdf/bc79906c3b98f546d7bc8bea98f61b105560763a.pdf.

OpenCompass Contributors. OpenCompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

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

Qwen Team. Qwen3-30B-A3B-Instruct-2507 model card. https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507, 2025.

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 *Proceedings of the Fifth International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Daniil Vankov, Nikita Ivkin, Jaime Campos Salas, Kyle R. Ulrich, Xiang Song, Ashish Khetan, and George Karypis. XShare: Collaborative in-Batch expert sharing for faster MoE inference. In *ICML 2026 Workshop AdaptFM*, 2026. URL https://openreview.net/forum?id=KmY5XUgEYe.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Juntong Wu, Jialiang Cheng, Fuyu Lv, Dan Ou, and Li Yuan. SERE: Similarity-based expert re-routing for efficient batch decoding in MoE models. In *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5d07eb2335d88b1dfcefcc3d2da2a230-Abstract-Conference.html.

Juntong Wu, Jialiang Cheng, Qishen Yin, Yue Dai, Yuliang Yan, Fuyu Lv, Ou Dan, and Li Yuan. BEAM: Binary expert activation masking for dynamic routing in MoE. *arXiv preprint arXiv:2605.14438*, 2026b. URL https://arxiv.org/abs/2605.14438.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*, 2024. URL https://arxiv.org/abs/2410.12013.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Seohee Yoon and Yong Suk Choi. COMPEL: Compensated mixture-of-experts pruning with expert-layer distribution. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30408–30423. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1521. URL https://aclanthology.org/2026.findings-acl.1521/.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. How well do large language models perform in arithmetic tasks? *arXiv preprint arXiv:2304.02015*, 2023. URL https://arxiv.org/abs/2304.02015.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. AdaMoE: Token-adaptive routing with null experts for mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6223–6235. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.361. URL https://aclanthology.org/2024.findings-emnlp.361/.

Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Zhiliang Wu, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, and Hehe Fan. Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15169–15186. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.820. URL https://aclanthology.org/2025.findings-emnlp.820/.

A Method and Implementation Details

A.1 DelMoE Pseudocode

Offline rectification coefficient selection. As defined in Section 3.2, we calibrate one rectification coefficient per model using three calibration tasks and the two configurations $(k, b) = (1, 0)$ and $(2, 0)$. Each candidate receives the equally weighted mean of the six configuration–task accuracy scores; the highest mean determines the selected rectification coefficient, with ties favoring the larger rectification coefficient. Appendix B specifies the calibration data and protocol.

The selected rectification coefficient is shared across all operating points, requests, layers, and generation steps for that model. Both prefill and decoding apply Algorithm A1 with this fixed rectification coefficient. At $s = 1$, route deletion still applies, while surviving weights retain their native values. The CUDA backend consumes the resulting routes and weights; it does not perform rectification coefficient selection.

Single-call routing and execution.

Algorithm A1 summarizes expert selection, route deletion, gate-weight rectification, and execution for a single batched MoE call.

Algorithm A1 DELMOE: selection, route deletion, gate-weight rectification, and execution

Require: Non-padding tokens $\mathcal{T}$ with hidden states h_t; native expert IDs $e_{t,j}$ and weights $g_{t,j} > 0$ for $1 \le j \le K$; backbone width $1 \le k \le K$; nonnegative integer budget b; offline-calibrated model-shared rectification coefficient $0 < s \le 1$

Ensure: Routed-branch outputs $\widetilde{y}_t$, retained routes $\mathcal{R}'$, and counts $U_{\mathrm{dec}}, N_{\mathrm{dec}}$

$P \leftarrow \bigcup_{t\in\mathcal{T}} \{e_{t,j} : 1 \le j \le k\}$
$C \leftarrow \left(\bigcup_{t\in\mathcal{T}} \{e_{t,j} : k < j \le K\}\right) \setminus P$
$d_t \leftarrow \mathbf{1}[\exists j > k : e_{t,j} \notin P]$ for every $t \in \mathcal{T}$
for each $e \in C$ **do**
    $n_e \leftarrow \sum_{t\in\mathcal{T}} \sum_{j=k+1}^{K} \mathbf{1}[e_{t,j} = e]$
    $M_e \leftarrow \sum_{t\in\mathcal{T}} \sum_{j=k+1}^{K} g_{t,j}\mathbf{1}[e_{t,j} = e]; \quad S(e) \leftarrow M_e^2/n_e$
end for
$A \leftarrow$ the $\min(b, |C|)$ highest-scoring experts in C; break ties by smaller ID
$H \leftarrow P \cup A; \quad \mathcal{R}' \leftarrow \{(t, e_{t,j}) : t \in \mathcal{T},\ 1 \le j \le K,\ e_{t,j} \in H\}$
for each native route $(t, e_{t,j})$ **do**
    if $e_{t,j} \notin H$ **then**
        $\widetilde{g}_{t,j} \leftarrow 0$; omit this route from execution
    else
        $\widetilde{g}_{t,j} \leftarrow s g_{t,j}$ if $d_t = 1$, and $g_{t,j}$ otherwise
    end if
end for
Group $\mathcal{R}'$ by expert, align each group, and execute only retained routes
Reduce $\widetilde{y}_t \leftarrow \sum_{j:e_{t,j}\in H} \widetilde{g}_{t,j} F_{e_{t,j}}(h_t)$ in native slot order
return $\widetilde{y}_t$, $\mathcal{R}'$, $U_{\mathrm{dec}} = |H|$, and $N_{\mathrm{dec}} = |\mathcal{R}'|$

A.2 VARIABLE-LENGTH CUDA BACKEND

The custom backend receives the $[|\mathcal{T}|, K]$ expert IDs and final gate weights produced by DELMOE. Each retained token–expert route becomes one physical assignment before alignment. A CUDA metadata kernel counts these assignments per expert, computes contiguous group offsets, stores their flattened token–slot indices in expert-grouped order, and records the expert ID for each aligned block. Groups are padded only to the operator's row-alignment granularity; all reported experiments use 16-row alignment, with $|\mathcal{T}|K$ as the sentinel index for padding positions.

The grouped fused-MoE kernels then use this metadata to execute the experts' fused gate/up projections and down projections, writing results to the original token–slot rows. Top-4 models reduce outputs in original slot order, whereas top-6 and top-8 models reuse vLLM's canonical MoE reduction. Metadata, activation, per-route output, and final-output buffers use graph-static capacities, preserving fixed kernel interfaces and buffer capacities under CUDA Graph replay. CUDA backend execution remains on the current GPU stream and requires neither host synchronization nor cross-batch state.

If expert e has m_e retained routes and the row-alignment granularity is a, its physical workload contains $a\lceil m_e/a \rceil$ rows; the total is $\sum_e a\lceil m_e/a \rceil$. Thus, $N_{\mathrm{dec}} = \sum_e m_e$ is the number of retained token–expert routes and therefore also the number of unpadded assignments, rather than the number of aligned execution rows. Let $\mathcal{E}$ denote the set of routed experts in the layer. For route selection and deletion, beyond the $O(|\mathcal{T}|K)$ route scan, maintaining the top supplemental candidates requires at most $O(|\mathcal{E}|b)$ comparisons and $O(|\mathcal{E}| + b)$ control space. These bounds exclude offline rectification coefficient calibration and expert FFN computation.

B EXPERIMENTAL SETUP

This appendix specifies the checkpoints, evaluation settings, and measurement protocols for the results in Section 4 and Appendix C.

Models, checkpoints, and accuracy configuration. Qwen1.5, ERNIE, and Qwen3 contain 24/27/48 MoE layers, 60/64/128 routed experts, hidden dimensions of 2048/2560/2048, expert intermediate dimensions of 1408/1536/768, and native routing widths $K = 4/6/8$, respectively; ERNIE's shared experts are excluded from the number of routed experts. We use fixed Hugging Face checkpoint revisions, including the Qwen3-30B-A3B-Instruct-2507 release (Qwen Team, 2025), and generation

settings (Table B1). Table B2 provides the task configurations for the eight-task evaluation in Section 4.1.

Table B1: Model revisions, routing widths, and OpenCompass generation settings.

Model	Native K	Temp.	Top-p	Top-k	Rep. penalty	Default max output
Qwen1.5-MoE-A2.7B-Chat	4	0.7	0.8	20	1.05	1024
ERNIE-4.5-21B-A3B-PT	6	0.8	0.8	50	1.00	1024
Qwen3-30B-A3B-Instruct-2507	8	0.7	0.8	20	1.00	2048

Model	*Hugging Face revision*
Qwen1.5	ec052fda178e241c7c443468d2fa1db6618996be
ERNIE	87db95487941cb39592ee0abca3b9155a6d19c5c
Qwen3	0d7cf23991f47feeb3a57ecb4c9cee8ea4a17bfe

Task configurations and scoring. We use OpenCompass 0.5.2, generation seed 0, and one evaluation worker. Table B2 lists the imported configurations, demonstrations, scoring rules, and effective output limits. We retain each configuration's evaluation split, prompt template, answer extraction, and evaluator; CMMLU and BoolQ use zero-shot chain-of-thought prompts. ERNIE and Qwen3 use model-level output limits only when a dataset specifies none. Three-task sweeps use the complete GSM8K, HumanEval, and MBPP evaluations from the eight-task protocol, without subsampling test items.

Table B2: OpenCompass task configurations. IDs are the six-character configuration suffixes; output limits are in tokens.

Task	Config ID	Demos	Scoring	Max output		
				Qwen1.5	ERNIE	Qwen3
CMMLU	`305931`	0	Option accuracy	1024	1024	2048
BoolQ	`1d56df`	0	Option accuracy	1024	1024	2048
BBH	`5b92b0`	3	Task-specific accuracy	1024	512	512
MATH	`cecb31`	4	MATH evaluator v2	1024	1024	1024
GSM8K	`1d7fe4`	4	Numeric-answer accuracy	1024	512	512
Math401	`ab5f39`	4	Numeric-answer accuracy	1024	512	512
HumanEval	`8e312c`	0	Pass@1	1024	512	512
MBPP	`1e1056`	3	Pass@1	1024	512	512

Serving runtime. All servers use Python 3.10.20 and BF16. Qwen1.5 and Qwen3 use vLLM 0.8.4 with PyTorch 2.6.0+cu124; ERNIE uses vLLM 0.10.2 with PyTorch 2.8.0+cu128. We use a maximum sequence length of 4,096 tokens, a GPU memory utilization target of 0.9, and TP $= 1$, with CUDA Graph execution enabled. The server uses its default maximum number of sequences, and the client imposes no concurrency cap.

Table B3: Hardware and core protocols for each experiment category.

Experiment	Scope and hardware	Core protocol	Metrics and checks
OpenCompass accuracy	All matched methods on three models; 1×A100	Fixed BS $= 16$; eight tasks; CoT for CMMLU/BoolQ; one complete evaluation per point	Task accuracy, *Avg.*
Routing profiles	All matched methods on three models; 1×A100	Fixed decode BS $= 16$; nearest-match measured U_{dec}	U_{dec}, N_{dec}
Multi-QPS serving	Three models; 1×A800 80GB; TP $= 1$	vLLM; Poisson; QPS $\in \{8, 16, 24\}$; exact 128/32 tokens; ignore EOS; 5,000 requests; fresh server; 256/480/720 warm-ups	Mean TPOT, Mean TTFT, completion and throughput checks
Qwen3 QPS $= 24$ repeats	13 settings × 3 prespecified repeats; 1×A800 80GB; TP $= 1$	Equal-weight three-run mean; min–max describes dispersion, not a confidence interval	Mean TPOT, Mean TTFT, min–max
Budget/backend studies	Three-model DELMOE operating points; 1×A800 80GB; TP $= 1$	QPS $= 16$; exact 128/32 tokens; 5,000 requests; identical routes and outputs for Compacted/Fixed-shape pairs	Mean TPOT, speedup
Controlled MoE-block microbenchmarks	Fig. 1(b): 13 pairs; Fig. 7(a): one pair; Qwen3; 1×A800; TP $= 1$	BS $= 16$; fixed hidden-state inputs; real router projection through routed-output reduction; CUDA Graph; 20 blocks × 200 replays; fresh process per case; no retries	Average MoE Time, components, within-run SD

Controlled MoE-block microbenchmarks. Under the protocol in Table B3, both microbenchmarks use Qwen3's real layer weights and router projection. Figure 1(b) reports a sweep of 13 paired configurations at $k = 1, b = 0$–12 (26 serial cases), with identical logical routes, gate weights, and mathematical outputs within each pair. Figure 7(a) uses an independent paired measurement

at $k = 1, b = 0$. CUDA events measure from router projection through routed-output reduction. Average MoE Time is the arithmetic mean of the 20 block means; their SD describes within-run dispersion, not independent-run uncertainty.

Multi-QPS statistics and controls. The multi-QPS measurements in Table B3 run serially. Requests follow a Poisson arrival process scheduled using absolute arrival times, with request and server seeds fixed to 0. Each request contains exactly 128 input tokens and generates 32 output tokens. Qwen1.5 and ERNIE have one complete measurement per setting, without automatic retries, outlier deletion, or selective method reruns. Qwen3/QPS $= 24$ uses three prespecified independent repeats for all 13 experimental settings; Figure 5 reports equal-weight means and min–max dispersion. Speedup uses Vanilla Mean TPOT from the same model–QPS group, including its three-run mean for Qwen3/QPS $= 24$.

Implementations and matching. Vanilla preserves native routing; Direct narrows the native width without constructing a batch-level expert set. Here K is the native width, and k is Direct's final width or the batch-aware methods' base width. The parameters ρ, m, b, s control SERE retention, XShare supplementation, DELMOE supplementation, and gate-weight calibration, respectively. Within the $k = 1$, $k = 2$, and Sat./Near-sat. groups, we match methods by nearest measured U_{dec} from fixed-BS16 A100 profiles. Sat. denotes the representative configurations identified in the budget study in Figure 6. For Qwen3, Table 1 instead reports multiple neighboring configurations in a near-saturation group; this grouping does not imply that every configuration retains over 99% of Vanilla accuracy. Profiles count only valid tokens and average U_{dec} and N_{dec} over MoE layers and decoding steps. Table 1 uses this unified matching protocol and the Qwen1.5 OEA-NR/XShare-NR adaptations justified in Appendix C.1; other models follow the original methods.

SERE similarity calibration. Our SERE baselines use model-specific expert-output similarities calibrated on 400 FineWeb-Edu (Penedo et al., 2024) training sequences of exactly 128 tokens, with seed 42 and each model's tokenizer. Qwen1.5 and Qwen3 use a shuffle buffer of 100,000; ERNIE draws from three fixed dataset shards without a shuffle buffer. The calibration microbatch sizes are 16, 1, and 1 for Qwen1.5, ERNIE, and Qwen3, respectively. For every MoE layer, we accumulate expert-output norm and Gram statistics across all calibration microbatches, compute global Frobenius distances, and form similarities as one minus the distance divided by the layer's maximum distance. The resulting matrices have shapes $24 \times 60 \times 60$, $27 \times 64 \times 64$, and $48 \times 128 \times 128$. ERNIE calibration covers routed experts only. These inputs and matrices are held fixed for the SERE comparisons; they are separate from DELMOE's rectification coefficient search. Appendix C.1 explains why these reproductions are not exact numerical replications of SERE's published table.

Offline calibration data and protocol. We use three calibration datasets, one for each task category: the GSM8K training split for mathematics, the BoolQ training split for general reasoning, and an MBPP-derived proxy calibration set for HumanEval-style code generation. This choice balances task coverage, sensitivity to the rectification coefficient, and the availability of calibration examples separate from final evaluation.

All three models use the same fixed calibration examples, with model-specific tokenizers and generation settings. The mathematics subset contains 512 examples from the GSM8K training split, with a fixed example order and batch size of 16. These examples are separate from the 1,319-example GSM8K test set. For BoolQ, we select 512 examples from the 9,427-example SuperGLUE (Wang et al., 2019) training split after excluding exact and near duplicates of validation and test examples. Selection uses proportional stratification by answer label and passage-length quartile, with a fixed hash-based ordering within each stratum. A further 128 examples are reserved for validation and excluded from coefficient calibration.

For code generation, we construct a HumanEval-style function-completion proxy from 256 MBPP training examples. We extract the function name, argument signature, and necessary code prefix from each reference solution, place the problem description in the docstring, and convert the first MBPP test assertion into a public input–output example in that docstring. The prompt begins with "Complete the following python code:" and asks the model to complete the function body; the reference function body is excluded. Generated completions are parsed into executable functions and scored using the remaining two MBPP assertions, which are hidden from the prompt, through HumanEval's restricted execution mechanism.

This proxy uses MBPP training problems rather than the 164 official HumanEval evaluation problems. It contains the same 256 underlying problems as the additional MBPP calibration subset, presented in a different format. We therefore include only the proxy in the coefficient-selection objective, avoiding double weighting of the same programming problems. The additional MBPP and MATH calibration results do not enter coefficient selection.

For each model, we evaluate each candidate rectification coefficient at $(k, b) = (1, 0)$ and $(2, 0)$ on the three calibration datasets. We average the six accuracy scores with equal weight and select the rectification coefficient with the highest mean, breaking ties in favor of the larger rectification coefficient. The selected rectification coefficient remains fixed throughout prefill and decoding. The selected values are 0.85, 0.95, and 0.95 for Qwen1.5, ERNIE, and Qwen3, respectively.

Table B4 reports the calibration mean for each candidate rectification coefficient.

Table B4: Offline rectification coefficient calibration. Each score (%) is the equally weighted mean of six accuracies: $(k, b) = (1, 0)$ and $(2, 0)$ on GSM8K train, BoolQ train, and the MBPP-derived HumanEval-style proxy. Bold indicates the highest mean for each model.

s	Qwen1.5	ERNIE	Qwen3
0.70	59.7656	84.0820	73.9583
0.75	61.9141	84.1146	74.2188
0.80	62.0117	84.8958	75.0000
0.85	**62.3698**	84.9935	75.1953
0.90	61.5885	84.7331	75.9766
0.95	61.5560	**85.2539**	**76.0091**
1.00	60.9049	85.0260	75.3255
Selected s	0.85	0.95	0.95

Rectification coefficient selection for the budget study. Tables C11–C13 list the fixed rectification coefficients and eight-task scores for the budget curves. Figure 6 selects rectification coefficients by the GSM8K, HumanEval, and MBPP mean on the common seven-value 0.70–1.00 grid (Appendix C.3); Qwen1.5 $(2, 0)$ therefore uses $s = 0.70$, whereas Table 1 uses the offline-calibrated model-shared rectification coefficient $s = 0.85$. These budget curves select rectification coefficients retrospectively and hold them fixed across tokens, layers, and decoding steps, so selection gains describe the evaluated grids rather than held-out task performance.

Table 4 instead compares Renorm, native weights, and the offline-calibrated model-shared rectification coefficients using eight-task means; Appendix C.5 provides the corresponding per-task scores.

C Additional Experimental Results

C.1 Baseline Reproduction and Compatibility Diagnostics

Published SERE results. SERE describes calibration on 400 FineWeb-Edu sequences of 128 tokens but does not release the exact samples, similarity matrices, or artifacts sufficient to reconstruct them. Table C1 retains its four published $\rho = 0$ points in the main table's $k = 1/k = 2$ regions. These scores are lower than the corresponding DelMoE $(k, b) = (1, 0)/(2, 0)$ scores, but serve only as cross-protocol context: SERE reports neither U_{dec} nor N_{dec} and names a different Qwen3 checkpoint. They are therefore excluded from Table 1's matching and rankings.

Table C1: Four $\rho = 0$ OpenCompass results reported in the SERE paper (Wu et al., 2026a). These cross-protocol values provide context only and are excluded from the matching and rankings in Table 1.

Model	Method (configuration)	CMMLU	BOOLQ	BBH	MATH	GSM8K	MATH_{401}	HEval	MBPP	*Avg.*↑
Qwen1.5	SERE $(k = 2, \rho = 0)$	68.12	80.15	33.16	14.06	50.19	58.35	46.95	26.20	47.15
	SERE $(k = 1, \rho = 0)$	60.09	79.85	32.71	7.58	33.36	52.12	17.07	20.20	37.87
Qwen3	SERE $(k = 2, \rho = 0)$	81.24	89.79	71.33	70.22	82.41	80.80	82.93	63.80	77.82
	SERE $(k = 1, \rho = 0)$	60.53	85.08	57.64	46.98	52.08	52.12	32.32	31.40	52.27

Qwen1.5 compatibility diagnostic. Qwen1.5 does not renormalize its native top-4 gate weights. Table C2's GSM8K/HumanEval/MBPP diagnostic shows that original selected-set renormalization severely reduces OEA and XShare accuracy relative to native top-4. Router traces show 3.32–$10.12\times$ amplification of the routed branch, identifying a mismatch with native gate semantics rather than a routing or execution error. OEA-NR/XShare-NR therefore retain the original expert-set construction and selected experts while preserving native router magnitudes. The eight-task run was stopped after diagnosis; its partial outputs are not reported as results, and the three-task diagnostic is excluded from Table 1's rankings.

Table C2: Three-task Qwen1.5 compatibility diagnostic for OEA and XShare with their original selected-set renormalization. This is not a complete eight-task evaluation.

Method	Configuration	GSM8K	HEval	MBPP	Three-task *Avg.*↑
Vanilla	$K = 4$	51.25	48.17	29.40	42.94
OEA	$k = 1$	1.74	0.00	0.00	0.58
OEA	$k = 2$	0.99	0.00	0.00	0.33
XShare	$k = 1, m = 0$	1.74	0.00	0.00	0.58
XShare	$k = 1, m = 1$	1.14	0.00	0.00	0.38
XShare	$k = 1, m = 2$	1.44	0.00	0.00	0.48
XShare	$k = 1, m = 3$	1.21	0.00	0.00	0.40

C.2 End-to-End Decoding Latency across Request Rates

Tables C3–C5 report all 117 experimental settings underlying Figure 5. Each numeric entry reports Mean TPOT / Mean TTFT in milliseconds, where TTFT denotes time to first token. Mean TPOT measures average per-output-token latency after the first token and is our primary decoding metric. Mean TTFT runs from request release to the first token and includes scheduling, queueing, and prefill, so we use it only as a first-token diagnostic. Vanilla depends only on model and QPS but is repeated across the three tables to support row-wise comparison; the $k = 1/k = 2$ tables use OEA, while Sat. uses XShare.

Table C3: Complete $k = 1$ multi-QPS serving results. Entries are Mean TPOT / Mean TTFT (ms).

Model	QPS	Vanilla	SERE	OEA	DELMOE Fixed	DELMOE Compacted
Qwen1.5	8	13.00/37.02	9.57/40.24	10.83/38.13	9.71/42.12	8.89/42.29
Qwen1.5	16	23.88/44.29	15.44/47.19	17.20/45.53	15.71/47.98	14.33/47.84
Qwen1.5	24	37.06/50.06	24.78/50.22	27.77/51.07	27.04/54.66	25.17/53.82
ERNIE	8	25.96/64.26	12.48/55.54	16.23/62.81	11.82/54.38	10.13/55.12
ERNIE	16	58.62/78.23	25.58/67.86	35.72/78.18	23.56/67.45	20.16/67.07
ERNIE	24	132.18/93.72	71.48/87.39	102.76/93.40	60.86/81.79	53.36/79.51
Qwen3	8	34.82/83.98	22.08/85.00	27.30/84.03	22.72/86.18	17.79/86.70
Qwen3	16	79.00/101.83	52.48/108.62	60.14/103.30	53.33/107.49	41.02/107.52
Qwen3	24	184.97/120.93	154.80/124.53	159.38/121.89	156.99/124.18	131.28/124.95

Table C4: Complete $k = 2$ multi-QPS serving results. Entries are Mean TPOT / Mean TTFT (ms).

Model	QPS	Vanilla	SERE	OEA	DELMOE Fixed	DELMOE Compacted
Qwen1.5	8	13.00/37.02	10.65/40.20	12.06/38.63	10.75/40.94	10.03/42.39
Qwen1.5	16	23.88/44.29	18.44/45.79	20.13/44.11	19.66/49.42	18.27/48.92
Qwen1.5	24	37.06/50.06	34.25/54.96	36.03/53.63	35.00/55.86	33.65/55.73
ERNIE	8	25.96/64.26	14.99/58.36	18.75/63.71	14.17/56.56	12.38/56.60
ERNIE	16	58.62/78.23	35.79/74.76	45.08/79.19	32.97/71.00	30.62/73.22
ERNIE	24	132.18/93.72	111.80/94.29	125.85/95.31	96.20/90.38	95.84/89.64
Qwen3	8	34.82/83.98	23.48/85.49	28.88/83.67	24.73/89.27	19.01/84.10
Qwen3	16	79.00/101.83	57.88/107.21	66.82/104.61	61.62/110.45	51.27/111.49
Qwen3	24	184.97/120.93	161.78/122.52	172.71/122.15	174.21/126.83	161.27/127.70

Table C5: Complete Sat. multi-QPS serving results. Entries are Mean TPOT / Mean TTFT (ms).

Model	QPS	Vanilla	SERE	XShare	DELMOE Fixed	DELMOE Compacted
Qwen1.5	8	13.00/37.02	10.29/40.48	13.70/38.80	11.38/41.43	10.62/41.49
Qwen1.5	16	23.88/44.29	16.55/45.56	21.12/45.08	18.55/47.74	17.73/48.74
Qwen1.5	24	37.06/50.06	29.97/54.85	31.90/50.27	31.50/55.35	31.61/57.65
ERNIE	8	25.96/64.26	13.21/56.12	19.22/63.88	13.24/54.87	11.74/55.30
ERNIE	16	58.62/78.23	28.35/71.41	40.79/77.48	27.69/69.95	23.55/66.93
ERNIE	24	132.18/93.72	67.45/83.72	108.89/94.55	68.02/84.43	62.13/82.44
Qwen3	8	34.82/83.98	24.06/85.84	33.46/85.22	25.53/85.83	21.73/88.99
Qwen3	16	79.00/101.83	58.23/106.02	74.79/105.62	65.39/112.25	53.71/108.52
Qwen3	24	184.97/120.93	168.23/123.62	171.55/120.11	173.60/126.80	164.99/126.75

The Mean TTFT change from Vanilla to DELMOE with its custom variable-length backend ranges from -15.2% to $+15.2\%$ and is strongly model-dependent: all nine ERNIE settings are lower, whereas all eighteen Qwen1.5 and Qwen3 settings are higher. We therefore make no claim of consistent first-token acceleration; the system claim remains scoped to post-first-token decoding TPOT. Moreover, every cell except Qwen3/QPS $= 24$ has one complete measurement, so the mean differences should not be interpreted as stable prefill or tail-latency effects.

All 143 server launches complete 5,000 measured requests with zero failures. Achieved request throughput spans 7.930–8.260, 15.831–16.514, and 23.660–24.753 requests/s at target QPS 8, 16, and 24, respectively. We do not duplicate ITL, E2E latency, or tail percentiles: ITL largely overlaps TPOT, fixed-length E2E latency mainly combines TTFT and TPOT, and single-run tail estimates would not support stable method-level inference.

Independent serving repeats. Table C6 gives the individual Qwen3/QPS $= 24$ measurements behind the means and ranges in Figure 5. Each run follows the same prespecified protocol in Appendix B. The run values and their means are rounded independently; all three runs contribute equally.

Table C6: Qwen3 at QPS 24: Mean TPOT (ms) from all three independent runs for each experimental setting. The final column is their equal-weight mean.

Setting	Method	Run 1	Run 2	Run 3	Mean
All	Vanilla	178.495	190.440	185.963	184.966
$k = 1$	SERE	162.841	160.019	141.531	154.797
$k = 1$	OEA	162.360	159.153	156.621	159.378
$k = 1$	DELMOE Fixed	153.720	156.530	160.713	156.988
$k = 1$	DELMOE Compacted	131.491	131.431	130.930	131.284
$k = 2$	SERE	158.519	159.082	167.753	161.785
$k = 2$	OEA	174.086	162.596	181.453	172.711
$k = 2$	DELMOE Fixed	186.680	157.428	178.513	174.207
$k = 2$	DELMOE Compacted	163.629	159.593	160.573	161.265
Sat.	SERE	173.019	170.227	161.433	168.227
Sat.	XShare	173.003	168.045	173.612	171.554
Sat.	DELMOE Fixed	189.186	157.760	173.867	173.604
Sat.	DELMOE Compacted	176.004	160.876	158.086	164.989

Batch-size diagnostic. Table C7 reports the number of active experts and all eight task scores for a decoding batch-size sweep of Qwen1.5 at $k = 1, b = 1, s = 0.85$, with all other settings fixed.

As the decoding batch grows from 4 to 32, U_{dec} rises from 4.81 to 22.49 and then remains near 22.5 through batch size 128, while the *8-task Avg.* gap to Vanilla narrows from 2.99 to 0.16 points. This diagnostic is consistent with later tokens increasingly reusing an already-saturated batch-level expert set; it is not presented as a cross-model scaling law.

Table C7: Qwen1.5 batch-size diagnostic ($k = 1, b = 1, s = 0.85$): mean U_{dec}, eight task scores, and their unweighted average. Vanilla provides the reference.

Batch	U_{dec}	CMMLU	BOOLQ	BBH	MATH	GSM8K	MATH401	HEval	MBPP	Avg.
Vanilla	–	69.14	80.12	35.09	13.92	51.25	59.85	48.17	29.40	48.37
4	4.81	68.09	79.94	35.47	10.74	43.29	56.86	39.02	29.60	45.38
8	8.16	68.67	79.85	35.88	10.12	42.76	58.85	41.46	27.60	45.65
16	13.95	69.03	79.97	36.05	10.88	46.17	57.86	39.02	27.60	45.82
32	22.49	69.17	79.85	36.40	11.60	45.87	58.10	41.46	26.00	46.06
64	22.61	69.02	79.72	36.52	13.04	48.90	60.85	46.34	31.20	48.20
128	22.53	69.19	79.72	35.82	14.04	52.01	58.35	45.73	30.80	48.21

C.3 Fine-Grained Batch-Aware Expert Selection

Figure 6 explores routing budgets using fixed rectification coefficients. For each of its 30 configurations, we select s from the same seven-value grid $\{0.70, 0.75, \ldots, 1.00\}$ by the equally weighted mean accuracy on GSM8K, HumanEval, and MBPP; exact ties favor the larger s. Tables C8–C10 report all 210 candidate means. Qwen1.5 $(1, 0)$–$(1, 4)$ and $(2, 0)$ use the three task columns of the eight-task coefficient sweeps; $(1, 5)$ and $(1, 6)$, ERNIE, and Qwen3 use the three-task coefficient sweeps. This retrospective selection characterizes the evaluated grid: the selection tasks also belong to the eight-task evaluation, so the curves do not measure held-out calibration performance or establish an upper bound outside this grid. Sat. denotes empirical saturation under this fixed rectification coefficient procedure.

Table C8: Qwen1.5: complete seven-value rectification coefficient scan for the Figure 6 budgets. Entries are the equally weighted mean of GSM8K, HumanEval, and MBPP; bold marks the selected maximum in each row.

(k, b)	$s = 0.70$	0.75	0.80	0.85	0.90	0.95	1.00
$(1, 0)$	34.99	**36.05**	34.25	34.59	33.28	31.88	33.50
$(1, 1)$	36.75	36.66	36.59	**37.60**	37.22	35.96	34.17
$(1, 2)$	38.87	39.56	37.92	**39.71**	38.01	38.51	38.07
$(1, 3)$	39.78	40.49	**42.41**	42.23	39.74	42.04	40.19
$(1, 4)$	41.29	**43.18**	40.03	42.50	41.95	42.15	39.57
$(1, 5)$	41.04	41.86	43.37	**44.98**	40.90	41.49	39.81
$(1, 6)$	42.03	43.03	43.49	**44.63**	41.99	41.46	40.67
$(2, 0)$	**42.88**	42.25	42.78	42.81	41.56	41.73	41.21

Table C9: ERNIE: complete seven-value rectification coefficient scan for the Figure 6 budgets. Entries are the equally weighted mean of GSM8K, HumanEval, and MBPP; bold marks the selected maximum in each row.

(k, b)	$s = 0.70$	0.75	0.80	0.85	0.90	0.95	1.00
$(1, 0)$	74.19	74.82	76.67	77.22	78.75	**78.98**	77.03
$(1, 1)$	75.94	78.10	78.50	78.48	78.19	**79.57**	77.92
$(1, 2)$	79.14	78.12	77.53	79.43	79.33	**80.23**	79.16
$(1, 3)$	77.63	78.11	79.22	78.82	79.31	79.59	**80.04**
$(2, 0)$	79.97	**81.33**	79.94	79.99	79.60	79.52	79.85

Table C10: Qwen3: complete seven-value rectification coefficient scan for the Figure 6 budgets. Entries are the equally weighted mean of GSM8K, HumanEval, and MBPP; bold marks the selected maximum in each row.

(k, b)	$s = 0.70$	0.75	0.80	0.85	0.90	0.95	1.00
$(1, 0)$	46.73	49.65	47.41	49.15	51.00	**51.61**	51.21
$(1, 1)$	59.58	62.03	**65.92**	64.39	62.78	63.23	65.89
$(1, 2)$	67.46	68.38	67.52	69.45	70.27	70.66	**71.78**
$(1, 3)$	71.92	71.14	70.68	**73.22**	72.28	72.58	71.57
$(1, 4)$	71.76	74.33	74.81	74.88	74.01	**75.36**	73.81
$(1, 5)$	74.84	75.22	76.23	76.74	**77.74**	75.73	77.26
$(1, 6)$	77.03	78.79	**79.17**	78.49	78.39	79.05	77.98
$(1, 7)$	79.36	80.31	**81.46**	80.49	79.69	80.42	78.65
$(1, 8)$	80.26	80.92	82.18	**82.54**	82.16	81.20	80.08
$(1, 9)$	80.83	81.65	82.62	82.55	**82.66**	81.50	79.88
$(1, 10)$	82.10	82.84	82.69	83.39	83.21	**83.39**	83.17
$(1, 11)$	83.00	**83.83**	83.15	83.38	83.43	82.54	81.64
$(1, 12)$	81.96	83.84	83.95	83.80	**84.57**	83.34	82.45
$(2, 0)$	77.85	80.28	79.34	**80.37**	79.19	80.36	79.85
$(2, 1)$	81.28	79.64	81.10	81.79	81.57	**82.17**	81.76
$(2, 2)$	81.24	81.72	**83.40**	82.45	82.91	83.30	82.97
$(2, 3)$	81.92	83.39	82.96	83.07	83.32	82.29	**83.45**

Tables C11–C13 report every routing-budget point and fixed rectification coefficient underlying Figure 6, together with all eight task scores, their unweighted average, and Mean TPOT from the independent QPS $= 16$ sweep. HEval abbreviates HumanEval; s is the fixed rectification coefficient selected for each point. On the common seven-value grid, Qwen1.5 $(2, 0)$ selects $s = 0.70$, with a three-task mean of 42.88 and an eight-task mean of 48.30; Table 1 instead uses the offline-calibrated model-shared rectification coefficient $s = 0.85$. For this Figure 6 point only, $U_{\mathrm{dec}} = 23.14$ and Mean TPOT $= 18.827$ ms are retained from the $s = 0.80$ measurements and marked with an asterisk. Changing s leaves the immediate route mask unchanged for a fixed layer input, but can alter later hidden states and routing during generation; these are reference coordinates, not new $s = 0.70$ measurements.

Table C11: Qwen1.5 routing-budget points in Figure 6. Eight-task accuracy uses BS $= 16$; Mean TPOT (ms) comes from a separate QPS $= 16$ serving sweep under the settings in Appendix B. Δ is the accuracy change from Vanilla (pp). *System measurements at $s = 0.80$ are retained as references for the $s = 0.70$ accuracy point.

(k, b)	s	Exam			Math			Code		*Avg.*↑	U_{dec} ↓	Δ	TPOT↓
		CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP				
Vanilla	—	69.14	80.12	35.09	13.92	51.25	59.85	48.17	29.40	48.37	38.25	+0.00	23.555
$(1, 0)$	0.75	67.44	77.98	35.24	8.32	42.53	57.11	38.41	27.20	44.28	12.94	−4.09	14.819
$(1, 1)$	0.85	69.03	79.97	36.05	10.88	46.17	57.86	39.02	27.60	45.82	13.95	−2.55	15.603
$(1, 2)$	0.85	69.32	79.69	35.84	11.12	46.85	57.11	43.29	29.00	46.53	14.99	−1.84	16.635
$(1, 3)$	0.80	69.37	79.82	35.93	11.98	48.07	58.10	47.56	31.60	47.80	15.88	−0.56	17.068
$(1, 4)$	0.75	69.24	79.48	35.86	11.64	49.58	59.85	48.17	31.80	48.20	16.97	−0.17	17.822
$(1, 5)$	0.85	69.71	79.63	36.28	12.60	50.87	60.10	54.27	29.80	49.16	17.95	+0.79	17.596
$(1, 6)$	0.85	69.65	79.63	36.43	12.32	51.25	58.60	52.44	30.20	48.82	18.94	+0.45	19.381
$(2, 0)$	0.70	67.55	78.84	37.37	12.38	49.28	61.60	48.17	31.20	48.30	23.14^{*}	−0.07	18.827^{*}

Measured U_{dec} distinguishes the nominal budget b from the realized number of active experts.

Table C12: ERNIE routing-budget points in Figure 6. Eight-task accuracy uses BS $= 16$; Mean TPOT (ms) comes from a separate QPS $= 16$ serving sweep under the settings in Appendix B. Δ is the accuracy change from Vanilla (pp).

(k, b)	s	Exam			Math			Code		*Avg.*↑	U_{dec} ↓	Δ	TPOT↓
		CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP				
Vanilla	—	80.99	86.88	80.09	70.40	90.60	78.30	79.88	69.60	79.59	41.80	+0.00	59.734
$(1, 0)$	0.95	81.14	86.24	74.71	67.90	89.08	78.55	79.27	68.60	78.19	12.33	-1.40	22.046
$(1, 1)$	0.95	81.28	86.36	76.53	70.40	89.23	77.81	80.49	68.80	78.86	13.37	-0.73	24.823
$(1, 2)$	0.95	81.42	86.76	77.77	70.58	90.60	81.55	79.88	70.20	79.84	14.42	+0.25	24.519
$(1, 3)$	1.00	81.26	87.00	78.87	70.02	90.45	80.05	79.88	69.80	79.67	15.39	+0.07	26.422
$(2, 0)$	0.75	81.11	86.09	76.02	70.46	90.67	78.55	81.71	71.60	79.53	21.14	-0.07	31.333

ERNIE's main-text representative remains $(k, b) = (1, 2)$. Although $(1, 3)$ also exceeds Vanilla accuracy, it does not improve the eight-task average over $(1, 2)$.

Table C13: Qwen3 routing-budget points in Figure 6. Eight-task accuracy uses BS $= 16$; Mean TPOT (ms) comes from a separate QPS $= 16$ serving sweep under the settings in Appendix B. Δ is the accuracy change from Vanilla (pp).

(k, b)	s	Exam			Math			Code		*Avg.*↑	U_{dec} ↓	Δ	TPOT↓
		CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP				
Vanilla	—	84.62	90.00	77.11	72.46	89.23	80.80	89.02	78.40	82.71	50.90	+0.00	84.993
$(1, 0)$	0.95	76.77	89.36	66.21	51.82	58.68	60.85	47.56	48.60	62.48	11.89	-20.22	47.602
$(1, 1)$	0.80	78.92	89.08	71.37	64.72	81.65	82.29	57.32	58.80	73.02	13.18	-9.69	43.800
$(1, 2)$	1.00	82.46	89.82	70.95	67.48	83.17	84.79	72.56	59.60	76.35	13.99	-6.35	53.503
$(1, 3)$	0.85	81.95	89.63	74.34	70.24	86.43	86.03	76.22	57.00	77.73	15.14	-4.98	51.018
$(1, 4)$	0.95	82.76	90.09	74.29	71.08	86.66	83.29	76.83	62.60	78.45	15.88	-4.26	52.177
$(1, 5)$	0.90	82.79	89.88	74.91	72.14	87.95	84.29	80.49	64.80	79.66	17.12	-3.05	56.596
$(1, 6)$	0.80	82.50	89.57	75.89	73.12	89.01	85.29	81.71	66.80	80.49	17.95	-2.22	51.217
$(1, 7)$	0.80	82.36	89.48	75.60	72.62	88.78	83.54	87.80	67.80	81.00	18.84	-1.71	54.950
$(1, 8)$	0.85	83.09	90.09	76.07	73.48	89.61	81.55	87.20	70.80	81.49	19.81	-1.22	58.338
$(1, 9)$	0.90	83.79	89.72	76.12	72.78	89.39	82.29	87.80	70.80	81.59	21.02	-1.12	57.671
$(1, 10)$	0.95	84.25	90.03	75.69	72.84	89.39	83.04	87.20	73.60	82.01	21.93	-0.70	58.494
$(1, 11)$	0.75	82.78	89.97	76.67	73.78	89.46	83.54	89.02	73.00	82.28	22.82	-0.42	61.499
$(1, 12)$	0.90	84.04	89.79	76.76	73.56	88.86	80.55	90.24	74.60	82.30	23.96	-0.40	61.617
$(2, 0)$	0.85	82.77	89.60	74.99	71.60	86.73	83.29	85.98	68.40	80.42	20.15	-2.29	49.792
$(2, 1)$	0.95	83.92	89.91	74.81	72.66	87.11	82.54	87.80	71.60	81.29	21.40	-1.41	53.761
$(2, 2)$	0.80	83.72	90.12	76.67	73.10	89.39	84.79	87.80	73.00	82.32	22.60	-0.38	60.251
$(2, 3)$	1.00	84.55	90.09	75.33	72.54	88.55	81.55	87.80	74.00	81.80	23.61	-0.90	60.684

Qwen3's $(2, 2)$ representative and its advantage over $(1, 11)$ at nearly equal U_{dec} are discussed in Section 4.4.1. Increasing the budget to $(2, 3)$ worsens both accuracy and latency, whereas $(1, 8)$ offers lower U_{dec} and TPOT at lower accuracy. The budget sweep and Figure 5 are separate measurement runs under the same protocol constraints, each with its own Vanilla reference.

Table C14 reports complete eight-task results for the matched selection comparisons in Table 2. It compares seven selection rules at five Qwen1.5 budgets ($k = 1, b = 1$–5) with $s = 1$, at Qwen1.5 $(1, 5)$ with $s = 0.85$, ERNIE $(k, b) = (1, 2)$, and Qwen3 $(2, 2)$, while retaining the earlier Qwen3 $(1, 8)$ comparison and supplemental controls. Each comparison fixes (k, b), the candidate expert set, and $s = 1$ unless explicitly marked otherwise. Table 2 shows four of these rules and bolds the best result among the displayed selectors; the full table below also includes M, M/n, and n.

Table C14: Selection rules with fixed candidate expert set C (eight tasks, batch size 16); s is fixed within each comparison.

Model	(k, b)	Selector	s	Exam			Math			Code		8-task	Δ8-task Avg.
				CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HumanEval	MBPP	Avg.↑	vs. Vanilla
Qwen1.5	$(1, 1)$	M^2/n (ours)	1.00	69.16	80.28	34.86	10.88	41.39	55.61	32.32	28.80	44.16	-4.21
Qwen1.5	$(1, 1)$	M	1.00	69.23	80.18	35.27	11.10	43.37	56.11	29.88	27.00	44.02	-4.35
Qwen1.5	$(1, 1)$	M/n	1.00	69.38	80.37	34.75	10.86	41.62	57.86	38.41	24.20	44.68	-3.68
Qwen1.5	$(1, 1)$	$\max(g)$	1.00	69.30	80.28	34.39	10.86	43.97	55.11	34.15	26.60	44.33	-4.03
Qwen1.5	$(1, 1)$	n	1.00	68.37	79.48	36.16	10.20	41.09	55.36	35.37	26.00	44.00	-4.36
Qwen1.5	$(1, 1)$	Fixed random	1.00	68.77	79.11	35.19	9.42	38.97	55.11	39.02	27.00	44.07	-4.29
Qwen1.5	$(1, 1)$	Reverse M^2/n	1.00	68.45	78.99	34.73	9.34	41.09	56.36	31.71	25.00	43.21	-5.16
Qwen1.5	$(1, 2)$	M^2/n (ours)	1.00	69.59	79.88	34.48	11.90	43.75	56.36	42.07	28.40	45.80	-2.56

Continued on next page

Table C14 (continued)

Model	(k, b)	Selector	s	Exam			Math			Code		8-task	Δ8-task Avg.
				CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HumanEval	MBPP	Avg.↑	vs. Vanilla
Qwen1.5	$(1, 2)$	M	1.00	69.68	80.21	35.38	11.74	44.35	59.35	36.59	28.60	45.74	-2.63
Qwen1.5	$(1, 2)$	M/n	1.00	69.36	80.21	34.55	11.48	43.44	56.11	38.41	28.60	45.27	-3.10
Qwen1.5	$(1, 2)$	$\max(g)$	1.00	69.36	80.40	35.23	11.26	44.12	57.36	34.76	28.80	45.16	-3.21
Qwen1.5	$(1, 2)$	n	1.00	68.97	79.72	35.68	10.64	43.97	55.61	31.10	27.60	44.16	-4.21
Qwen1.5	$(1, 2)$	Fixed random	1.00	69.30	78.75	35.30	9.94	39.65	55.86	37.20	25.80	43.98	-4.39
Qwen1.5	$(1, 2)$	Reverse M^2/n	1.00	69.18	79.08	35.81	8.82	39.12	53.87	34.15	27.40	43.43	-4.94
Qwen1.5	$(1, 3)$	M^2/n (ours)	1.00	69.36	79.72	35.05	11.94	47.08	59.10	43.29	30.20	46.97	-1.40
Qwen1.5	$(1, 3)$	n	1.00	69.24	80.09	35.88	9.98	43.44	59.85	32.93	28.20	44.95	-3.42
Qwen1.5	$(1, 3)$	M/n	1.00	69.50	80.03	34.49	11.82	44.58	59.10	38.41	27.00	45.62	-2.75
Qwen1.5	$(1, 3)$	M	1.00	69.27	80.00	35.05	12.84	44.50	58.10	39.02	29.00	45.97	-2.40
Qwen1.5	$(1, 3)$	$\max(g)$	1.00	69.53	80.28	35.24	12.78	44.28	58.10	34.15	29.20	45.45	-2.92
Qwen1.5	$(1, 3)$	Fixed random	1.00	69.12	79.02	35.77	9.68	42.68	55.61	35.98	28.20	44.51	-3.86
Qwen1.5	$(1, 3)$	Reverse M^2/n	1.00	69.26	78.90	35.04	9.90	39.12	55.11	35.37	25.20	43.49	-4.88
Qwen1.5	$(1, 4)$	M^2/n (ours)	1.00	69.20	80.03	35.43	12.68	46.63	58.85	42.07	30.00	46.86	-1.51
Qwen1.5	$(1, 4)$	M	1.00	69.15	80.15	35.34	12.50	45.79	58.60	38.41	31.60	46.44	-1.92
Qwen1.5	$(1, 4)$	M/n	1.00	69.25	80.03	34.79	12.52	46.02	60.35	39.63	29.60	46.52	-1.84
Qwen1.5	$(1, 4)$	$\max(g)$	1.00	69.37	79.94	35.05	12.54	46.10	59.35	47.56	29.80	47.46	-0.90
Qwen1.5	$(1, 4)$	n	1.00	69.23	79.85	35.69	11.36	42.68	58.10	38.41	27.40	45.34	-3.03
Qwen1.5	$(1, 4)$	Fixed random	1.00	69.07	79.17	35.32	9.96	41.47	57.36	35.37	26.40	44.27	-4.10
Qwen1.5	$(1, 4)$	Reverse M^2/n	1.00	68.85	79.14	35.85	9.98	39.42	56.86	30.49	26.20	43.35	-5.02
Qwen1.5	$(1, 5)$	M^2/n (ours)	1.00	69.22	80.03	34.41	13.50	48.98	59.60	41.46	29.00	47.03	-1.34
Qwen1.5	$(1, 5)$	M	1.00	69.25	80.03	35.16	12.78	48.60	56.36	42.68	31.20	47.01	-1.36
Qwen1.5	$(1, 5)$	M/n	1.00	69.13	79.76	34.47	12.60	48.29	57.36	43.90	29.60	46.89	-1.48
Qwen1.5	$(1, 5)$	$\max(g)$	1.00	69.14	79.91	34.88	12.62	47.54	57.36	45.73	31.40	47.32	-1.04
Qwen1.5	$(1, 5)$	n	1.00	69.20	79.97	35.35	11.94	46.25	57.36	40.85	29.80	46.34	-2.03
Qwen1.5	$(1, 5)$	Fixed random	1.00	69.10	79.60	34.98	9.86	44.20	56.11	34.15	26.20	44.28	-4.09
Qwen1.5	$(1, 5)$	Reverse M^2/n	1.00	69.06	79.02	35.93	9.94	40.03	53.12	32.32	25.60	43.13	-5.24
Qwen1.5	$(1, 3)$	M^2/n (ours)	0.80	69.37	79.82	35.93	11.98	48.07	58.10	47.56	31.60	47.80	-0.56
Qwen1.5	$(1, 5)$	M^2/n (ours)	0.85	69.71	79.63	36.28	12.60	50.87	60.10	54.27	29.80	49.16	$+0.79$
Qwen1.5	$(1, 5)$	M	0.85	69.66	79.63	36.38	12.14	48.52	59.10	48.17	30.80	48.05	-0.32
Qwen1.5	$(1, 5)$	M/n	0.85	69.66	79.69	36.10	13.02	49.20	59.85	49.39	30.60	48.44	$+0.07$
Qwen1.5	$(1, 5)$	$\max(g)$	0.85	69.57	79.79	36.41	12.22	48.90	59.60	46.34	30.00	47.85	-0.51
Qwen1.5	$(1, 5)$	n	0.85	69.67	79.51	36.78	10.62	46.78	59.60	40.85	29.80	46.70	-1.66
Qwen1.5	$(1, 5)$	Fixed random	0.85	69.59	79.45	36.52	9.00	43.75	58.35	41.46	29.00	45.89	-2.48
Qwen1.5	$(1, 5)$	Reverse M^2/n	0.85	69.39	79.02	36.11	9.52	42.15	57.11	37.20	26.40	44.61	-3.76
ERNIE	$(1, 2)$	M^2/n (ours)	1.00	81.16	86.18	77.86	70.46	90.83	76.31	78.66	68.00	78.68	-0.91
ERNIE	$(1, 2)$	M	1.00	81.16	86.70	78.22	69.62	90.22	78.80	79.88	68.80	79.17	-0.42
ERNIE	$(1, 2)$	M/n	1.00	81.13	86.61	77.25	69.26	89.92	78.55	74.39	69.00	78.26	-1.33
ERNIE	$(1, 2)$	$\max(g)$	1.00	80.71	87.19	77.38	70.48	90.83	81.05	79.27	69.80	79.59	0.00
ERNIE	$(1, 2)$	n	1.00	80.79	86.27	77.49	70.44	89.61	79.80	78.05	70.40	79.11	-0.48
ERNIE	$(1, 2)$	Fixed random	1.00	81.04	85.96	76.04	68.78	89.23	76.06	75.00	68.40	77.57	-2.02
ERNIE	$(1, 2)$	Reverse M^2/n	1.00	81.44	85.66	75.12	67.94	89.84	77.56	77.44	67.80	77.85	-1.74
Qwen3	$(1, 8)$	M^2/n (ours)	1.00	83.94	90.18	75.02	72.88	88.70	81.80	82.93	68.60	80.51	-2.20
Qwen3	$(1, 8)$	M	1.00	83.86	90.24	73.75	69.74	86.58	84.04	78.66	57.60	78.06	-4.65
Qwen3	$(2, 2)$	M^2/n (ours)	1.00	84.47	89.60	74.73	72.40	87.72	81.05	87.80	73.40	81.40	-1.31
Qwen3	$(2, 2)$	M	1.00	84.24	90.15	74.31	72.24	87.41	81.30	85.37	71.80	80.85	-1.86
Qwen3	$(2, 2)$	M/n	1.00	84.27	90.28	75.01	72.60	88.63	80.55	85.98	74.80	81.51	-1.20
Qwen3	$(2, 2)$	$\max(g)$	1.00	84.40	90.31	75.81	72.84	88.32	82.54	86.59	74.80	81.95	-0.76
Qwen3	$(2, 2)$	n	1.00	84.18	89.82	74.44	72.18	86.13	81.80	86.59	71.40	80.81	-1.90
Qwen3	$(2, 2)$	Fixed random	1.00	84.14	90.03	74.07	71.70	85.90	82.04	85.37	70.20	80.43	-2.28
Qwen3	$(2, 2)$	Reverse M^2/n	1.00	83.91	90.37	74.10	72.04	86.05	83.29	85.98	68.20	80.49	-2.22

C.4 Route Deletion vs. Route Replacement

Table 3 summarizes the route-action comparison. All rows use BS $= 16$, $b = 0$, $s = 1$, and the batch union of native top-k experts ($k = 1, 2$), without M^2/n supplementation. Only out-of-set route actions vary: source-route weights remain unchanged, and Expert reroute uses $\rho = 0$. The comparison isolates route actions rather than complete baseline methods.

Tables C15–C17 provide the corresponding eight-task scores and means for each model.

Table C15: Qwen1.5: route-action ablation on a shared top-k backbone ($b = 0$, $s = 1$, BS$= 16$). Scores are percentages; Avg. is the unweighted eight-task mean. HEval denotes HumanEval. Task scores and means are rounded independently.

k	Route action	CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP	Avg.
1	Route deletion	67.77	79.66	35.18	9.20	41.39	53.87	31.71	27.40	43.27
1	Expert share	67.83	78.87	33.99	7.34	37.00	55.11	23.78	23.60	40.94
1	Expert reroute	46.93	78.41	30.71	4.54	22.29	46.63	3.66	12.40	30.70
2	Route deletion	68.96	80.09	34.85	13.88	50.11	59.10	45.73	27.80	47.57
2	Expert share	68.82	79.33	35.35	13.50	48.22	56.86	42.68	27.80	46.57
2	Expert reroute	67.34	80.37	33.82	12.48	47.99	54.11	38.41	26.80	45.17

Table C16: ERNIE: route-action ablation on a shared top-k backbone ($b = 0$, $s = 1$, BS$= 16$). Scores are percentages; Avg. is the unweighted eight-task mean. HEval denotes HumanEval. Task scores and means are rounded independently.

k	Route action	CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP	Avg.
1	Route deletion	81.09	86.06	74.98	68.44	89.92	77.81	73.17	68.00	77.43
1	Expert share	80.98	86.51	76.25	66.66	87.19	72.57	84.15	69.20	77.94
1	Expert reroute	79.60	85.14	69.49	56.80	81.50	66.33	71.34	59.20	71.18
2	Route deletion	81.33	86.21	78.62	70.60	90.67	81.30	79.27	69.60	79.70
2	Expert share	81.10	86.76	80.09	72.08	90.67	80.05	78.66	70.00	79.93
2	Expert reroute	80.82	86.09	78.79	70.26	90.67	78.30	80.49	69.40	79.35

Table C17: Qwen3: route-action ablation on a shared top-k backbone ($b = 0$, $s = 1$, BS$= 16$). Scores are percentages; Avg. is the unweighted eight-task mean. HEval denotes HumanEval. Task scores and means are rounded independently.

k	Route action	CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP	Avg.
1	Route deletion	77.52	89.60	66.21	52.62	56.63	58.10	49.39	47.60	62.21
1	Expert share	72.82	87.25	66.52	43.40	57.01	55.86	44.51	51.80	59.90
1	Expert reroute	51.41	73.98	40.09	28.72	39.35	42.89	17.07	15.80	38.66
2	Route deletion	83.82	90.09	74.48	71.30	84.99	81.80	85.98	68.60	80.13
2	Expert share	83.70	90.18	74.09	69.94	86.58	81.80	84.15	68.20	79.83
2	Expert reroute	78.37	89.02	67.41	64.96	79.23	74.31	81.10	55.00	73.68

C.5 Gate-Weight Rectification after Route Deletion

Tables C18–C20 provide the eight task scores underlying Table 4, comparing Renorm, native weights ($s = 1$), and the offline-calibrated model-shared rectification coefficient at the same nine operating points. The shared rectification coefficients are 0.85, 0.95, and 0.95 for Qwen1.5, ERNIE, and Qwen3, respectively. HEval denotes HumanEval; task scores and averages are rounded independently.

Table C18: Qwen1.5: per-task accuracy for the three gate-weight rules in Table 4 (BS$= 16$). Avg. is the unweighted eight-task mean.

(k, b)	Rule	s	CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP	Avg.
$(1, 0)$	Renorm	—	55.44	78.75	22.94	2.66	7.13	29.68	0.61	4.60	25.23
$(1, 0)$	native	1.00	67.77	79.66	35.18	9.20	41.39	53.87	31.71	27.40	43.27
$(1, 0)$	Model-shared	0.85	67.65	78.17	36.69	8.90	40.18	55.11	35.98	27.60	43.79
$(2, 0)$	Renorm	—	69.07	80.15	30.91	11.72	41.62	53.87	29.27	24.20	42.60
$(2, 0)$	native	1.00	68.96	80.09	34.85	13.88	50.11	59.10	45.73	27.80	47.57
$(2, 0)$	Model-shared	0.85	68.65	79.51	35.87	13.88	51.10	60.35	45.12	32.20	48.34
$(1, 5)$	Renorm	—	69.24	80.03	31.15	7.40	24.64	50.12	21.34	21.20	38.14
$(1, 5)$	native	1.00	69.22	80.03	34.41	13.50	48.98	59.60	41.46	29.00	47.03
$(1, 5)$	Model-shared	0.85	69.71	79.63	36.28	12.60	50.87	60.10	54.27	29.80	49.16

Table C19: ERNIE: per-task accuracy for the three gate-weight rules in Table 4 (BS= 16). Avg. is the unweighted eight-task mean.

(k, b)	Rule	s	CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP	Avg.
$(1, 0)$	Renorm	—	75.78	83.18	65.68	53.40	78.09	72.57	60.98	45.80	66.93
$(1, 0)$	native	1.00	81.09	86.06	74.98	68.44	89.92	77.81	73.17	68.00	77.43
$(1, 0)$	Model-shared	0.95	81.14	86.24	74.71	67.90	89.08	78.55	79.27	68.60	78.19
$(2, 0)$	Renorm	—	80.37	86.97	78.32	63.50	88.63	78.55	76.22	64.60	77.14
$(2, 0)$	native	1.00	81.33	86.21	78.62	70.60	90.67	81.30	79.27	69.60	79.70
$(2, 0)$	Model-shared	0.95	81.04	86.27	78.21	71.42	90.30	79.55	78.66	69.60	79.38
$(1, 2)$	Renorm	—	78.96	85.20	73.87	59.74	87.79	71.82	66.46	52.00	71.98
$(1, 2)$	native	1.00	81.16	86.18	77.86	70.46	90.83	76.31	78.66	68.00	78.68
$(1, 2)$	Model-shared	0.95	81.42	86.76	77.77	70.58	90.60	81.55	79.88	70.20	79.84

Table C20: Qwen3: per-task accuracy for the three gate-weight rules in Table 4 (BS= 16). Avg. is the unweighted eight-task mean.

(k, b)	Rule	s	CMMLU	BoolQ	BBH	MATH	GSM8K	Math401	HEval	MBPP	Avg.
$(1, 0)$	Renorm	—	43.01	80.09	14.51	1.26	4.47	18.45	5.49	5.00	21.54
$(1, 0)$	native	1.00	77.52	89.60	66.21	52.62	56.63	58.10	49.39	47.60	62.21
$(1, 0)$	Model-shared	0.95	76.77	89.36	66.21	51.82	58.68	60.85	47.56	48.60	62.48
$(2, 0)$	Renorm	—	71.46	89.91	55.68	47.76	62.40	66.08	78.05	47.00	64.79
$(2, 0)$	native	1.00	83.82	90.09	74.48	71.30	84.99	81.80	85.98	68.60	80.13
$(2, 0)$	Model-shared	0.95	83.56	90.12	74.09	71.92	87.11	81.05	85.37	68.60	80.23
$(2, 2)$	Renorm	—	77.85	90.58	64.41	58.86	76.95	77.56	80.49	66.40	74.14
$(2, 2)$	native	1.00	84.47	89.60	74.73	72.40	87.72	81.05	87.80	73.40	81.40
$(2, 2)$	Model-shared	0.95	83.94	90.03	75.21	72.26	88.32	81.80	85.98	72.60	81.27

Task sensitivity to the rectification coefficient. Figure C1 reports exploratory sensitivity to the rectification coefficient on the evaluation-task coefficient sweeps, rather than scores on the final calibration datasets. For each task, mean regret is the mean, over sampled operating points, of the difference between the best score on that point's coefficient grid and the score at the displayed coefficient. The figure covers five ERNIE and seventeen Qwen3 operating points on $s \in \{0.70, 0.75, \ldots, 1.00\}$. Across the evaluation-task coefficient sweeps, HumanEval has a larger mean within-point score range than MBPP: 9.75 vs. 3.43 percentage points for Qwen1.5, 6.46 vs. 2.80 for ERNIE, and 8.75 vs. 3.45 for Qwen3. The Qwen1.5 summary uses seven operating points on $s \in \{0.65, 0.70, \ldots, 1.05\}$. These observations motivate emphasizing HumanEval-style code generation in calibration; they do not establish that the MBPP-derived proxy preserves HumanEval's sensitivity or determine the final model-shared coefficients.

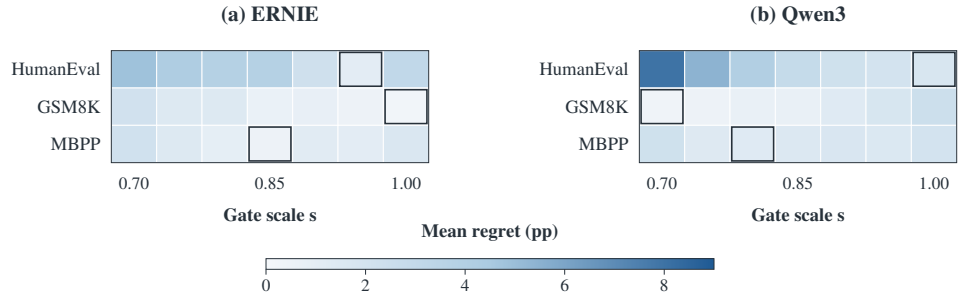

Figure C1: Per-task mean regret (pp) for ERNIE and Qwen3. Outlines mark minima on each task's evaluated grid. These are evaluation-task sweeps, not the final three-task calibration results.

C.6 Variable-Length vs. Fixed-Shape Execution

Controlled MoE-block timings. Table C21 reports all 13 paired configurations in Figure 1(b). Within each pair, logical routes, gate weights, and mathematical output are held fixed. Fixed-shape execution

processes $N_{\text{exec}} = 128$ slots before alignment, whereas compacted execution uses $N_{\text{exec}} = N_{\text{dec}}$. The two implementations therefore have the same U_{dec} and N_{dec} but different physical workloads. Table C22 separately reports the independent case in Figure 7(a), including its profiling breakdown. Its $b = 0$ measurement is not substituted for the $b = 0$ row of the sweep. Both tables report within-run block SDs, as defined in Appendix B, rather than uncertainty across independent runs.

Table C21: Complete Figure 1(b) sweep on Qwen3 at batch size 16 and $k = 1$. Latency is mean $\pm$ within-run SD (μs); reduction is relative to fixed-shape execution.

b	s	U_{dec}	N_{dec}	Fixed-shape	Compacted	Reduction
0	0.95	12	45	208.475 ± 0.147	141.653 ± 0.100	32.1%
1	0.80	13	50	221.982 ± 0.115	163.438 ± 0.069	26.4%
2	1.00	14	51	232.754 ± 0.147	174.593 ± 0.079	25.0%
3	0.85	15	54	245.310 ± 0.092	184.501 ± 0.100	24.8%
4	0.95	16	56	257.671 ± 0.083	197.686 ± 0.147	23.3%
5	0.90	17	59	267.655 ± 0.090	206.764 ± 0.138	22.7%
6	0.80	18	61	276.128 ± 0.099	215.914 ± 0.099	21.8%
7	0.80	19	64	283.337 ± 0.086	222.989 ± 0.172	21.3%
8	0.85	20	66	305.067 ± 0.255	229.678 ± 0.079	24.7%
9	0.90	21	68	315.361 ± 0.200	249.517 ± 0.202	20.9%
10	0.95	22	70	325.715 ± 0.236	258.614 ± 0.226	20.6%
11	0.75	23	72	332.145 ± 0.174	266.361 ± 0.218	19.8%
12	0.90	24	74	341.360 ± 0.275	271.214 ± 0.262	20.5%

Table C22: Independent Figure 7(a) timing breakdown (μs) for Qwen3 at batch size 16, $k = 1, b = 0$, $U_{\text{dec}} = 12$, and $N_{\text{dec}} = 45$.

Component	Fixed-shape	Compacted
Router projection	27.812	28.177
Routing after projection	92.237	24.592
Expert FFNs	80.598	79.757
Activation and reduction	7.604	8.696
Other	0.000	0.566
Total (mean $\pm$ within-run SD)	208.250 ± 0.085	141.788 ± 0.149