

Boundary-Aware Expert Ordering Preservation with Next-Router Supervision for MoE Quantization

Hancheol Park^{1*} Geonho Lee^{2†} Tae-Ho Kim^{1*}

¹Nota Inc. ²Hanyang University

¹{hancheol.park, thkim}@nota.ai, ²thisisho@hanyang.ac.kr

Abstract

Mixture-of-Experts (MoE) language models scale model capacity through sparse expert activation, but their large parameter footprint remains costly to deploy. Post-training quantization (PTQ) offers a practical compression strategy; however, quantization errors can alter router decisions and cause the quantized model to activate experts different from those selected by its full-precision counterpart. Prior router-aware PTQ objectives mitigate such expert-shift by aligning router outputs or preserving top- k expert orderings and margins at a given router. Yet sequential block-wise PTQ introduces an additional failure mode: quantizing a decoder block changes the representations received by the next MoE router, potentially causing downstream expert-shift that is not directly optimized by existing objectives. To address this issue, we propose *MENDS-MoE* (Margin-Preserving Expert Orderings with Next-Router Decision Supervision), an auxiliary calibration objective that supervises current-block quantization using routing decisions at the next MoE router. *MENDS-MoE* further introduces a boundary-aware margin loss that preserves orderings among selected experts while maintaining their separation from unselected experts near the top- k boundary. It is applied only during calibration and adds no inference-time computation. Across three MoE language models under 4-bit and 3-bit-average weight-only quantization, *MENDS-MoE* achieves the best average downstream accuracy and the lowest perplexity in most evaluated model-bit settings. Ablation studies further support the contributions of both next-router supervision and boundary-aware margin preservation.

1 Introduction

Mixture-of-Experts (MoE) language models scale total parameter capacity without a proportional in-

crease in per-token computation by routing each token to only a small subset of experts (Shazeer et al., 2017; Fedus et al., 2022). However, deploying these models still requires storing the parameters of all experts, making post-training quantization (PTQ) attractive for reducing their memory footprint. MoE quantization, in turn, introduces an additional failure mode: small quantization-induced perturbations in router scores can change the selected top- k experts. Following Chen et al. (2025a), we refer to this change in expert selection as *expert-shift*.

Recent router-aware PTQ methods mitigate expert-shift by explicitly preserving routing behavior under quantization. EAC-MoE and EAQuant align router outputs or routing distributions between full-precision and quantized models (Chen et al., 2025a; Fu et al., 2025), while ExpertQuant preserves ordered top- k selections and score margins among selected experts (Fang and Huang, 2025). These methods improve routing fidelity at their target router, but do not explicitly account for routing changes at the following MoE block caused by quantizing the current decoder block.

This missing supervision is particularly relevant to practical LLM PTQ pipelines. To keep calibration memory and computation tractable, PTQ methods typically optimize quantization sequentially at the layer or decoder-block level rather than jointly over the entire model. GPTQ performs layer-wise quantization based on local reconstruction (Frantar et al., 2023), whereas OmniQuant and AutoRound—the PTQ backbone used in our experiments—optimize quantization parameters through block-wise reconstruction (Shao et al., 2024; Cheng et al., 2024). EAC-MoE similarly adopts a layer-by-layer quantization and router-calibration pipeline for MoE models (Chen et al., 2025a). Under this sequential paradigm, quantizing a decoder block perturbs the representation passed to the next MoE router; thus, a block can be recon-

*Corresponding authors.

†Work done during an internship at Nota Inc.

structed accurately while still changing subsequent expert selections.

Instead, one might minimize the discrepancy between the hidden representations of the full-precision and quantized models over the entire network. However, even such global output matching does not guarantee downstream routing consistency, because a small residual difference in the representation fed to a router can reverse expert rankings near the top- k boundary. For example, consider a next-router decision with $k = 2$, where experts with scores 1.50 and 1.01 are selected, while an unselected expert receives a score of 1.00. A small perturbation caused by quantizing the preceding block can reduce the second selected score to 1.00 and increase the unselected score to 1.01. Although the score changes are small, the selected expert set changes. This illustrates why low reconstruction error alone does not guarantee downstream routing preservation.

To address this limitation, we propose MENDS-MoE (Margin-Preserving Expert Orderings with Next-Router Decision Supervision), an auxiliary objective for block-wise MoE quantization. MENDS-MoE supervises current-block quantization using routing decisions at the next MoE router, directly targeting downstream expert-shift induced by block quantization. It further introduces a boundary-aware margin loss that preserves both orderings among selected experts and margins between selected experts and unselected experts near the top- k boundary. Because it is applied only during calibration, MENDS-MoE introduces no additional inference-time computation.

Across three MoE language models under 4-bit and 3-bit-average weight-only quantization, MENDS-MoE generally improves the performance of quantized MoE models across downstream and language-modeling evaluations. Ablation studies further show that both next-router supervision and boundary-aware margin preservation contribute to these improvements.

Contributions. Our contributions are threefold: (1) we introduce next-router decision supervision to optimize current-block quantization for downstream routing; (2) we propose a boundary-aware margin loss that extends expert-ordering preservation to selected–unselected pairs near the top- k boundary; and (3) across three MoE language models and two low-bit settings, we show that MENDS-MoE achieves the best perplexity and av-

erage downstream accuracy among compared PTQ methods in most model–precision settings.

2 Related Work

Quantization of MoE language models. Early MoE-specific PTQ studies primarily address expert-level quantization challenges caused by sparse and heterogeneous expert activation. For example, QuantMoE analyzes the sensitivity of different MoE components to quantization, while MoEQuant improves calibration through expert-balanced sampling and affinity-guided quantization (Li et al., 2024; Chen et al., 2025b). More recently, preserving routing behavior has emerged as an important direction for MoE PTQ. EAC-MoE shows that replacing quantization-perturbed expert selections with the full-precision routing behavior substantially recovers the performance of quantized MoE models, demonstrating that expert-shift is a major source of quantization degradation (Chen et al., 2025a). Our work follows this router-alignment direction.

Router-alignment objectives for MoE quantization. Recent MoE PTQ methods add routing-aware losses to calibration objectives for optimizing quantization-related variables, such as quantization scale factors, zero-points, clipping thresholds, or rounding parameters. EAC-MoE’s TopK-MSE matches router logits for high-probability experts between the full-precision and quantized models (Chen et al., 2025a). Since its alignment range may exceed the routed expert count k , we denote it as *TopM-MSE*, where M is the number of aligned experts. Thus, TopM-MSE can include high-probability experts outside the routed set, but aligns their logit values through MSE. EAQuant aligns router logits and routing distributions (Fu et al., 2025), whereas ExpertQuant preserves ordered top- k selections and score gaps among selected experts (Fang and Huang, 2025).

Relation to our work. MENDS-MoE differs in two aspects. First, it supervises current-block quantization with decisions at the next MoE router, directly addressing downstream routing perturbations induced by block-wise PTQ. Second, rather than matching router-output values or preserving margins only among selected experts, its boundary-aware margin loss directly preserves ordering margins between selected experts and unselected experts near the top- k boundary. These margins pro-

test the expert membership relations most vulnerable to quantization-induced reversal.

3 Proposed Method

We propose MENDS-MoE, an auxiliary calibration objective for block-wise MoE quantization. Rather than replacing the underlying quantizer, MENDS-MoE augments its reconstruction objective with routing-aware terms. The method has two components: a boundary-aware expert-ordering loss and a next-router decision supervision loss. The former preserves routing margins that determine expert selection, while the latter guides the quantization of a decoder block using its effect on routing in the following MoE block.

3.1 Preliminaries

Weight-only post-training quantization. Let $\mathbf{W}$ be a full-precision weight tensor and let $\mathbf{W}_q = Q_b(\mathbf{W}; \theta)$ be its b -bit quantized counterpart, where θ denotes quantization-related variables such as scale factors, zero-points, clipping thresholds, or rounding parameters. In calibration-based PTQ, these variables are selected or optimized on a small calibration set to reconstruct the output hidden states of each quantized decoder block with respect to its full-precision counterpart. Our experiments use weight-only quantization, and MENDS-MoE adds no parameters to the inference-time model.

Sparse routing in MoE blocks. For the representation $\mathbf{x}_{\ell,t}$ of token t at an MoE block ℓ , a router produces logits $\mathbf{r}_{\ell}^{(t)} = R_{\ell}(\mathbf{x}_{\ell,t}) \in \mathbb{R}^E$ over E routed experts. The routed expert set is

$$\mathcal{T}_{\ell}(t) = \text{TopK}(\mathbf{r}_{\ell}^{(t)}, k). \quad (1)$$

The routed-expert contribution is then computed as

$$\mathbf{y}_{\ell,t}^{\text{routed}} = \sum_{e \in \mathcal{T}_{\ell}(t)} g_{\ell,e}^{(t)} E_{\ell,e}(\mathbf{x}_{\ell,t}), \quad (2)$$

where $g_{\ell,e}^{(t)}$ is the mixing weight of selected expert e , obtained from the router logits through a normalization function, such as softmax or normalized affinity scores. Architectures with shared experts may additionally include their always-active outputs in the MoE computation. Consequently, a small quantization-induced perturbation in router logits can replace a routed expert in $\mathcal{T}_{\ell}(t)$ and alter the computation path for the token.

3.2 Overall Objective

We quantize decoder blocks sequentially. Let $h_{\ell,t}^{\text{orig}}$ and $h_{\ell,t}^q$ be the output representations of block ℓ for token t in the full-precision and quantized paths, respectively. A standard block-wise PTQ objective minimizes their reconstruction error:

$$\mathcal{L}_{\text{rec}}^{(t)} = \left\| h_{\ell,t}^q - h_{\ell,t}^{\text{orig}} \right\|_2^2, \quad (3)$$

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \mathcal{L}_{\text{rec}}^{(t)}, \quad (4)$$

where $\mathcal{B}$ denotes the tokens in a calibration mini-batch.

MENDS-MoE supplements this objective with a current-router loss $\mathcal{L}_{\text{router}}$ and a downstream-router loss $\mathcal{L}_{\text{dr}}$. Because not every block produces a meaningful downstream routing disturbance, the latter is enabled only when a pilot quantization of block ℓ sufficiently changes the top- k selection at the next router. The resulting per-block objective is

$$\mathcal{L}_{\ell} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{router}} + \mathbb{I}[\sigma_{\ell \rightarrow \ell+1} < \tau] \lambda_2 \mathcal{L}_{\text{dr}}. \quad (5)$$

Here, $\sigma_{\ell \rightarrow \ell+1}$ measures the average overlap between the top- k experts selected by the next router when receiving the full-precision output of block ℓ and those selected when receiving its initially quantized output; τ is the threshold below which the downstream loss is activated. $\mathcal{L}_{\text{router}}$ is included only if block ℓ contains an MoE router; $\mathcal{L}_{\text{dr}}$ is included only if a subsequent MoE router is available. For the final decoder block, $\mathcal{L}_{\text{dr}} = 0$.

3.3 Boundary-Aware Expert-Ordering Loss

From logit matching to ordering preservation.

A router-aware calibration baseline is TopM-MSE, adapted from the TopK-MSE objective of EAC-MoE. Let $\mathbf{r}_{\text{orig}} \in \mathbb{R}^E$ and $\mathbf{r}_q \in \mathbb{R}^E$ be the router logits in the full-precision and quantized paths. For each token, TopM-MSE matches logits at the m expert indices ranked highest by the full-precision router:

$$\mathcal{L}_{\text{TopM}} = \frac{1}{m} \sum_{e \in \mathcal{M}} (r_q(e) - r_{\text{orig}}(e))^2, \quad (6)$$

where $\mathcal{M}$ contains the top- m full-precision expert indices and $m \geq k$. We use the notation TopM-MSE because the alignment range m may be larger than the number k of experts actually routed for a token.

Although TopM-MSE can include experts outside the routed set, it matches logit values rather than directly protecting expert orderings. For example, for $k = 1$ and $\mathbf{r}_{\text{orig}} = [1.0, 0.0]$, both $\mathbf{r}_q^{(A)} = [2.0, 1.0]$ and $\mathbf{r}_q^{(B)} = [0.0, 1.0]$ have squared error 2 from the original logits. Yet the first preserves the selected expert, whereas the second reverses the ranking and selects a different expert. This example motivates preserving the pairwise margins induced by the full-precision router, so that neighboring experts in its ranking retain their relative order after quantization, especially near the top- k selection boundary.

Expert pairs near the routing boundary. For token t , let $\pi_t(a)$ be the expert at rank a under the full-precision router, with rank 0 denoting the largest logit. We preserve two types of local ordering relations. First, adjacent selected-expert pairs retain the order within the routed set:

$$\mathcal{P}_{\text{adj}}(t) = \{(\pi_t(a), \pi_t(a+1)) : 0 \leq a < k-1\}. \quad (7)$$

Second, we construct a boundary pool from the lowest-ranked selected experts and the closest unselected experts:

$$\mathcal{I}_t = \{\pi_t(a) : k - b_{\text{sel}} \leq a \leq k + b_{\text{out}} - 1\}. \quad (8)$$

All ordered pairs inside this local pool are included:

$$\mathcal{P}_{\text{bnd}}(t) = \{(\pi_t(a), \pi_t(b)) : a < b, \pi_t(a), \pi_t(b) \in \mathcal{I}_t\}. \quad (9)$$

In particular, this set contains selected-unselected pairs around the top- k cutoff, whose reversal changes expert membership. The full pair set is

$$\mathcal{P}(t) = \mathcal{P}_{\text{adj}}(t) \cup \mathcal{P}_{\text{bnd}}(t). \quad (10)$$

For each $(i, j) \in \mathcal{P}(t)$, define the full-precision and quantized pairwise margins as

$$m_{\text{orig}}^{(t)}(i, j) = r_{\text{orig}}^{(t)}(i) - r_{\text{orig}}^{(t)}(j), \quad (11)$$

$$m_q^{(t)}(i, j) = r_q^{(t)}(i) - r_q^{(t)}(j). \quad (12)$$

We penalize violations of a relaxed reference margin:

$$z_{ij}^{(t)} = \eta m_{\text{orig}}^{(t)}(i, j) - m_q^{(t)}(i, j), \quad (13)$$

where $\eta \in (0, 1]$ controls the fraction of the full-precision margin to be retained. The token-level loss is

$$\mathcal{L}_{\text{router}}^{(t)} = \frac{1}{|\mathcal{P}(t)|} \sum_{(i,j) \in \mathcal{P}(t)} \text{softplus}(z_{ij}^{(t)}), \quad (14)$$

and the mini-batch loss is

$$\mathcal{L}_{\text{router}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \mathcal{L}_{\text{router}}^{(t)}. \quad (15)$$

Unlike exact logit matching, this objective allows harmless logit changes as long as the quantized router maintains the required ordering margins.

3.4 Next-Router Decision Supervision

Block reconstruction can produce similar output errors under different quantization parameters, while those outputs may lead to different decisions at the next MoE router. We therefore use downstream routing behavior to guide the quantization of block ℓ .

The current-router and next-router objectives play complementary roles. The former preserves the routing decision made within the block being quantized, whereas the latter captures how quantization-induced changes in the block output affect the routing decision of the following MoE block. Thus, next-router supervision extends routing preservation beyond local block reconstruction without requiring joint optimization of multiple blocks.

Let $A_{\ell+1}^{\text{orig}}$ be the full-precision prefix of the next block up to its MoE router, including operations before routing, and let $R_{\ell+1}^{\text{orig}}$ denote its fixed router. Feeding the full-precision and quantized outputs of block ℓ through this reference path gives

$$\mathbf{r}_{\ell+1}^{\text{orig}} = R_{\ell+1}^{\text{orig}}(A_{\ell+1}^{\text{orig}}(h_{\ell}^{\text{orig}})), \quad (16)$$

$$\mathbf{r}_{\ell+1}^q = R_{\ell+1}^{\text{orig}}(A_{\ell+1}^{\text{orig}}(h_{\ell}^q)). \quad (17)$$

The parameters of $A_{\ell+1}^{\text{orig}}$ and $R_{\ell+1}^{\text{orig}}$ remain frozen. Thus, gradients from downstream routing supervision update only the quantization parameters of the current block.

We construct $\mathcal{P}_{\ell+1}(t)$ from the ranking of $\mathbf{r}_{\ell+1}^{\text{orig}}(t)$ using the same boundary-aware pair construction as in section 3.3. The corresponding margin violation is

$$z_{\ell+1,ij}^{(t)} = \eta m_{\ell+1,\text{orig}}^{(t)}(i, j) - m_{\ell+1,q}^{(t)}(i, j), \quad (18)$$

and the downstream losses are

$$\mathcal{L}_{\text{dr}}^{(t)} = \frac{1}{|\mathcal{P}_{\ell+1}(t)|} \sum_{(i,j) \in \mathcal{P}_{\ell+1}(t)} \text{softplus}(z_{\ell+1,ij}^{(t)}), \quad (19)$$

$$\mathcal{L}_{\text{dr}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \mathcal{L}_{\text{dr}}^{(t)}. \quad (20)$$

3.5 Adaptive Activation with an RTN Pilot

We empirically observe that applying the downstream router loss to every block with an available next MoE router substantially degrades quantized-model performance (Appendix A). This observation suggests that next-router supervision should be applied selectively, rather than uniformly imposed on blocks whose downstream routing is already sufficiently preserved.

To determine whether block ℓ requires downstream supervision, we need to estimate how its quantization affects routing at the next MoE router before optimizing its trainable quantization parameters. Since the final quantized output of block ℓ is not yet available at this stage, we use round-to-nearest (RTN) quantization as a low-cost pilot approximation. In particular, only the current block ℓ is temporarily quantized with RTN, while the downstream prefix and router remain fixed. Thus, the pilot isolates the downstream routing perturbation caused by quantizing the current block.

Let $h_{\ell,t}^{\text{RTN}}$ denote the output of the RTN-quantized pilot for block ℓ at token t . Passing this representation through the fixed prefix and router of the next MoE block gives

$$\mathbf{r}_{\ell+1}^{\text{RTN}} = R_{\ell+1}^{\text{orig}} \left(A_{\ell+1}^{\text{orig}} (h_{\ell,t}^{\text{RTN}}) \right). \quad (21)$$

The corresponding reference logits $\mathbf{r}_{\ell+1}^{\text{orig}}$ are obtained using the output of the full-precision current block under the same downstream reference path. Hence, the two paths differ only in whether the current block ℓ is kept in full precision or temporarily quantized with RTN.

For each token, we compare the top- k expert sets selected by the next router under the reference and RTN-pilot paths:

$$\begin{aligned} \mathcal{T}_{\text{orig}}(t) &= \text{TopK}(\mathbf{r}_{\ell+1}^{\text{orig}}(t), k), \\ \mathcal{T}_{\text{RTN}}(t) &= \text{TopK}(\mathbf{r}_{\ell+1}^{\text{RTN}}(t), k). \end{aligned} \quad (22)$$

We then define the downstream routing overlap of block ℓ as

$$\sigma_{\ell \rightarrow \ell+1} = \frac{1}{N_t k} \sum_{t=1}^{N_t} |\mathcal{T}_{\text{orig}}(t) \cap \mathcal{T}_{\text{RTN}}(t)|, \quad (23)$$

where N_t is the number of probe tokens. By construction, $\sigma_{\ell \rightarrow \ell+1} \in [0, 1]$. A large overlap indicates that quantizing the current block is unlikely to change the next router’s selected expert set,

whereas a small overlap indicates a larger downstream routing perturbation. Accordingly, we activate the downstream loss only when

$$\mathbb{I}_{\text{dr}}(\ell) = \mathbb{I}[\sigma_{\ell \rightarrow \ell+1} < \tau], \quad (24)$$

where τ is the overlap threshold.

Concretely, we quantize MoE blocks sequentially. Before optimizing each block, we construct an RTN pilot for the current block alone and compute its overlap with the reference path using 32 calibration sequences of length 2,048, yielding $N_t = 65,536$ probe tokens. When $k = 6$, each token contributes the fraction of six reference-selected experts that remain selected under the RTN pilot. For example, for the first MoE block, we obtain $\sigma_{0 \rightarrow 1} = 0.9407$. Since this value is above $\tau = 0.865$, $\mathcal{L}_{\text{dr}}$ is disabled while optimizing this block, and quantization proceeds to the next block.

4 Experiments

4.1 Experimental Setup

Models. We evaluate MENDS-MoE on three open MoE language models: DeepSeek-V2-Lite-Chat (DeepSeek-AI, 2024), Moonlight-16B-A3B-Instruct (Liu et al., 2025), and Qwen3-30B-A3B (Yang et al., 2025). These models span different MoE architectures and parameter scales, allowing us to assess whether the proposed routing-aware objective transfers across diverse MoE backbones. We abbreviate their names in the result tables for readability.

Quantization settings. We implement MENDS-MoE on top of AutoRound, a learning-based block-wise PTQ framework (Cheng et al., 2024; Intel, 2026). All methods use weight-only symmetric quantization with group size 128. We consider two settings: standard 4-bit quantization and a more challenging 3-bit-average setting following EAC-MoE (Chen et al., 2025a). In the latter setting, experts in the latter half of the model are quantized to 2 bits, yielding an average model precision of approximately 3 bits. All compared methods use the same quantization configuration, calibration set, and optimization budget. Additional implementation details and hyperparameters are provided in appendix B.

Calibration data. We construct the calibration set from Nemotron-Post-Training-Dataset-v1 (NVIDIA, 2025). Specifically, we randomly sample 512 examples from its chat, code, math, and

Table 1: Performance under 4-bit weight-only quantization. For each model, the best method per benchmark (excluding Original) is in **bold**.

Model	Method	PPL ↓	Accuracy ↑					
		WikiText-2	MMLU-Pro	GSM8K-CoT	GPQA-D	HumanEval	IFEval	Avg.
DeepSeek-V2-Lite	Original	8.02	27.75	57.77	25.76	57.32	44.18	42.55
	AutoRound	8.37	25.24	56.41	23.74	51.83	40.30	39.50
	TopM-MSE	8.38	23.35	58.07	18.18	53.05	38.26	38.18
	ExpertQuant	8.74	19.53	42.76	23.74	45.12	36.23	33.48
	MENDS-MoE	8.30	25.24	55.88	24.75	53.66	43.07	40.52
Moonlight-16B-A3B	Original	8.46	37.01	76.72	30.81	73.17	44.73	52.49
	AutoRound	9.20	35.64	74.53	27.78	69.51	41.77	49.85
	TopM-MSE	9.16	33.49	75.13	26.26	69.51	40.67	49.01
	ExpertQuant	9.36	34.38	73.84	26.77	67.07	41.22	48.66
	MENDS-MoE	9.13	34.57	75.66	30.30	69.51	42.14	50.44
Qwen3-30B-A3B	Original	8.70	70.46	90.90	45.45	92.07	84.47	76.67
	AutoRound	9.04	68.78	90.07	36.36	92.07	82.81	74.02
	TopM-MSE	9.11	68.99	89.69	34.34	90.85	82.62	73.30
	ExpertQuant	9.40	68.02	89.92	40.40	90.24	82.62	74.24
	MENDS-MoE	9.08	69.44	90.30	41.41	92.07	82.99	75.24

STEM subsets and use sequences of length 2,048. This mixture provides calibration inputs covering instruction following, programming, mathematical reasoning, and scientific content.

Baselines. We compare MENDS-MoE with three baselines instantiated under the same AutoRound backbone. AutoRound uses only its standard block-reconstruction objective. TopM-MSE augments AutoRound with the router-logit matching loss adapted from EAC-MoE. Because the number of aligned router logits can exceed the number k of routed experts, we denote this baseline as TopM-MSE and set $M = 2k$ to include high-probability unselected experts near the routing boundary. ExpertQuant augments AutoRound with the Rank-Aware Jaccard (RAJ) and Gap Hinge (GH) losses proposed for rank-aware router alignment (Fang and Huang, 2025).

Controlled comparison. Because all routing-aware baselines are instantiated under the same AutoRound backbone, our comparison isolates the effect of the routing objective rather than differences in the underlying quantization pipeline. This controlled setup is particularly suitable for evaluating MENDS-MoE, whose contribution is an auxiliary calibration objective rather than a new quantization framework. It also ensures that observed performance differences are attributable to the routing-aware calibration objectives under a matched quantization setting.

Evaluation. We evaluate language modeling quality using WikiText-2 perplexity (PPL) (Merity et al., 2017), following the evaluation protocol commonly used in LLM quantization studies (Frantar et al., 2023). We further evaluate downstream performance on five benchmarks: MMLU-Pro (Wang et al., 2024), GSM8K with chain-of-thought prompting (Cobbe et al., 2021), GPQA-Diamond (Rein et al., 2024), HumanEval (Chen et al., 2021), and IFEval (Zhou et al., 2023). Downstream evaluations are conducted with lm-evaluation-harness (EleutherAI, 2026) using a vLLM backend (Kwon et al., 2023). We report the average accuracy over the five downstream benchmarks. Detailed prompting and generation settings are given in Appendix B.

4.2 Results

Overall performance. Tables 1 and 2 present the main results. MENDS-MoE achieves the highest average downstream accuracy on all three models under 4-bit quantization and on two of the three models under the more challenging 3-bit-average setting. For language modeling, MENDS-MoE obtains the lowest WikiText-2 perplexity on two models under 4-bit quantization and on all three models under 3-bit-average quantization. These results show that MENDS-MoE improves quantized MoE models across both language-modeling and downstream evaluations.

Table 2: Performance under 3-bit-average weight-only quantization. For each model, the best method per benchmark (excluding Original) is in **bold**.

Model	Method	PPL ↓	Accuracy ↑					
		WikiText-2	MMLU-Pro	GSM8K-CoT	GPQA-D	HumanEval	IFEval	Avg.
DeepSeek-V2-Lite	Original	8.02	27.75	57.77	25.76	57.32	44.18	42.55
	AutoRound	10.76	23.60	48.98	20.20	37.20	37.71	33.54
	TopM-MSE	10.80	22.07	50.27	23.74	36.59	36.60	33.85
	ExpertQuant	11.33	16.13	36.39	22.73	31.10	33.83	28.03
	MENDS-MoE	10.72	21.27	45.19	25.25	38.41	36.78	33.38
Moonlight-16B-A3B	Original	8.46	37.01	76.72	30.81	73.17	44.73	52.49
	AutoRound	15.58	31.52	60.20	20.71	39.02	39.37	38.16
	TopM-MSE	15.46	32.80	59.67	23.74	43.29	39.56	39.81
	ExpertQuant	16.35	31.07	53.60	23.23	45.73	37.71	38.27
	MENDS-MoE	13.93	31.09	60.73	22.73	48.17	40.11	40.57
Qwen3-30B-A3B	Original	8.70	70.46	90.90	45.45	92.07	84.47	76.67
	AutoRound	11.48	64.67	89.08	31.31	84.15	75.79	69.00
	TopM-MSE	11.53	64.15	87.87	28.28	85.37	75.60	68.25
	ExpertQuant	12.05	61.86	88.02	30.30	83.54	71.16	66.98
	MENDS-MoE	11.29	63.93	89.16	32.32	87.20	75.97	69.72

Task-level behavior. MENDS-MoE achieves the best or tied-best HumanEval accuracy in all six model–precision configurations, and the best GPQA-Diamond and IFEval accuracy in five settings each. It also leads on GSM8K-CoT in four settings, whereas its gains on MMLU-Pro are less consistent. This pattern suggests that MENDS-MoE is particularly effective for code generation and instruction following, with additional gains on several reasoning-oriented evaluations. Understanding why downstream routing preservation benefits some task types more consistently than others requires further task- and token-level analysis, which we leave for future work.

5 Discussion

Effect of next-router decision supervision. As shown in Table 3, retaining $\mathcal{L}_{\text{dr}}$ yields lower or tied perplexity in five of the six model–precision settings and higher average downstream accuracy in four settings. Its effect is particularly clear in the 3-bit-average setting. On Moonlight-16B-A3B, removing $\mathcal{L}_{\text{dr}}$ increases PPL from 13.93 to 15.84 and reduces average accuracy from 40.57 to 38.89. On Qwen3-30B-A3B, retaining $\mathcal{L}_{\text{dr}}$ reduces PPL from 11.62 to 11.29 and improves average accuracy from 61.04 to 69.72. Although the loss does not improve every metric in every setting, these results show that next-router supervision generally improves quantized-model performance, especially under stronger quantization.

Figure 1 provides a routing-level view of this

effect under 3-bit-average quantization; the corresponding overlap analysis under 4-bit quantization is provided in Appendix C. The full method and its $-\mathcal{L}_{\text{dr}}$ ablation use the same random seed and optimize the same objective until downstream supervision is first activated; their overlap curves therefore coincide in the earlier blocks. After activation, MENDS-MoE generally preserves higher overlap with the full-precision next-router expert sets, with the effect most visible for DeepSeek-V2-Lite and Moonlight-16B-A3B, where 3-bit-average quantization induces larger routing perturbations than in Qwen3-30B-A3B. For Qwen3-30B-A3B, the effect is less uniform across individual blocks, but the full method still recovers higher overlap at multiple activated blocks. This analysis confirms that $\mathcal{L}_{\text{dr}}$ improves the downstream routing behavior it is designed to preserve.

Effect of boundary-aware margin preservation.

Removing boundary pairs tests whether preserving non-boundary ordering relations alone is sufficient. Retaining selected–unselected boundary pairs reduces PPL in five of the six model–precision settings and improves average downstream accuracy in four settings. The benefit is particularly clear on Qwen3-30B-A3B: boundary-aware margin preservation improves average accuracy from 72.34 to 75.24 at 4-bit and from 68.66 to 69.72 in the 3-bit-average setting. It also lowers PPL on both DeepSeek-V2-Lite settings and both Qwen3-30B-A3B settings. The main exception is Moonlight-

Table 3: Ablation of MENDS-MoE components. Each “—” denotes an *independent* ablation that removes the indicated component from the full method (the ablations are *not* cumulative): “ $-\mathcal{L}_{dr}$ ” drops the downstream-router loss, and “— boundary pair” drops the boundary-pair selection mechanism. PPL is WikiText-2 ($\downarrow$); Avg is the unweighted mean accuracy ($\uparrow$) across the five tasks in our experiments. The best value per (Model, Bit) row is in **bold**.

Model	Bit	Method	PPL $\downarrow$	Avg $\uparrow$
DeepSeek	4	MENDS-MoE	8.30	40.52
		— $\mathcal{L}_{dr}$	8.31	39.97
		— boundary pair	8.32	39.99
	3	MENDS-MoE	10.72	33.38
		— $\mathcal{L}_{dr}$	10.72	34.27
		— boundary pair	10.78	33.91
Moonlight	4	MENDS-MoE	9.13	50.44
		— $\mathcal{L}_{dr}$	9.16	51.52
		— boundary pair	9.29	49.61
	3	MENDS-MoE	13.93	40.57
		— $\mathcal{L}_{dr}$	15.84	38.89
		— boundary pair	13.73	40.75
Qwen3	4	MENDS-MoE	9.08	75.24
		— $\mathcal{L}_{dr}$	9.06	74.31
		— boundary pair	9.11	72.34
	3	MENDS-MoE	11.29	69.72
		— $\mathcal{L}_{dr}$	11.62	61.04
		— boundary pair	11.31	68.66

16B-A3B under 3-bit-average quantization, where removing boundary pairs slightly improves both PPL and average accuracy. Overall, these results suggest that explicitly preserving selected–unselected margins near the top- k boundary is generally beneficial, although its contribution varies with the model and quantization severity.

Summary of the ablation findings. Taken together, the two components of MENDS-MoE target complementary sources of routing error: next-router supervision addresses routing perturbations propagated from the currently quantized block, while boundary-aware margin preservation protects expert membership near the top- k cutoff. Their combined use provides strong overall performance, with their relative contributions varying across models and precision settings.

6 Conclusion

We presented MENDS-MoE, an auxiliary calibration objective for post-training quantization of Mixture-of-Experts language models. The method targets a limitation of sequential block-wise PTQ: quantizing a decoder block can alter expert selec-

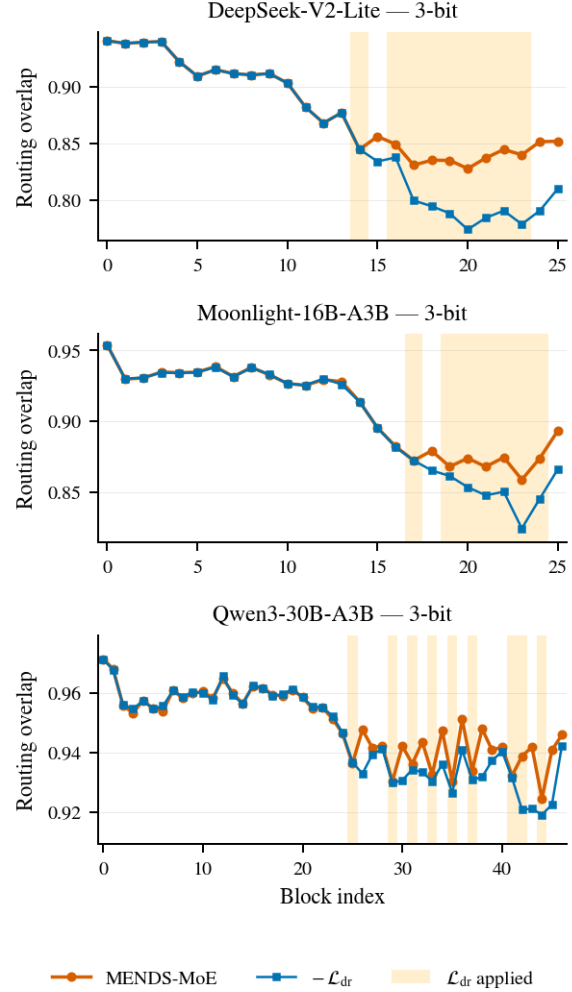


Figure 1: Next-router top- k overlap in the 3-bit-average setting, averaged over 65,536 probe tokens. Shaded regions indicate blocks selected for downstream supervision.

tion at the next MoE router even when local reconstruction error remains small. To address this issue, MENDS-MoE combines next-router decision supervision with a boundary-aware margin objective that preserves expert orderings within the routed set and across the top- k selection boundary. Experiments on three MoE language models under 4-bit and 3-bit-average weight-only quantization show that the proposed objective improves both language-modeling perplexity and downstream accuracy across most evaluated settings, without adding inference-time computation. Our analyses further indicate that downstream supervision and boundary-aware margin preservation are generally useful components, while adaptive activation avoids over-regularizing blocks whose downstream routing is already stable.

Limitations

Our experiments evaluate MENDS-MoE on three open MoE language models using weight-only quantization with AutoRound as the underlying PTQ framework. Although the method is formulated as an auxiliary calibration objective and is therefore compatible in principle with other learning-based quantizers, we do not yet establish its effectiveness with additional PTQ backbones, activation quantization, or lower-precision numerical formats beyond the settings considered in this work.

MENDS-MoE also introduces calibration-time choices, including loss weights, boundary-pool size, and the overlap threshold used to activate downstream supervision. While it incurs no additional inference-time computation, determining these settings requires validation during calibration. Moreover, the ablation results show that routing preservation is not uniformly beneficial for every metric and model–precision setting: next-router supervision or boundary-aware margin preservation can occasionally slightly degrade a particular evaluation measure. A more detailed analysis of when routing preservation aligns with downstream task quality remains an important direction for future work.

Finally, our objective treats the routing behavior of the full-precision model as the reference to be preserved. It therefore aims to retain the behavior of the original model under quantization, rather than correcting weaknesses, biases, or undesirable behaviors already present in that model.

Ethical Considerations

Mixture-of-Experts language models can be expensive to deploy because their expert parameters must be stored even though only a subset is activated for each token. By improving the fidelity of low-bit MoE quantization, MENDS-MoE may reduce memory requirements and serving costs for deploying existing models on more limited hardware. This may support broader access to efficient language-model inference and reduce the resources required for some deployment settings.

At the same time, making capable language models less expensive to serve can also lower barriers to harmful or irresponsible use. Our method does not add new capabilities, training data, external tools, or inference-time mechanisms; it is designed to better preserve the behavior of an existing full-

precision MoE model after quantization. Consequently, any safety risks, biases, privacy concerns, or misuse potential of the original model may remain present in its quantized version. Models compressed with MENDS-MoE should therefore undergo the same safety evaluation, access control, and deployment review as their full-precision counterparts.

Acknowledgments

This work was supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

References

- 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, and 39 others. 2021. [Evaluating large language models trained on code](#). *arXiv preprint arXiv:2107.03374*.
- Yuanteng Chen, Yuantian Shao, Peisong Wang, and Jian Cheng. 2025a. [EAC-MoE: Expert-selection aware compressor for mixture-of-experts large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12942–12963, Vienna, Austria.
- Zhixuan Chen, Xing Hu, Dawei Yang, Zukang Xu, Xu Chen, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. 2025b. [MoEQuant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 8245–8260. PMLR.
- Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, Lv Kaokao, and Yi Liu. 2024. [Optimize weight rounding via signed gradient descent for the quantization of LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11332–11350, Miami, Florida, USA.
- 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. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- DeepSeek-AI. 2024. [DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model](#). *arXiv preprint arXiv:2405.04434*.

- EleutherAI. 2026. [The language model evaluation harness](#). Version v0.4.11. Software release on Zenodo.
- Yi-Zeng Fang and Juinn-Dar Huang. 2025. [Router choice matters: Rank-aware post-training quantization for MoE models](#). OpenReview. ICLR 2026 Withdrawn Submission.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#). *Journal of Machine Learning Research*, 23(120):1–39.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2023. [OPTQ: Accurate quantization for generative pre-trained transformers](#). In *The Eleventh International Conference on Learning Representations*.
- Zhongqian Fu, Ning Ding, Kai Han, Xianzhi Yu, Xiaosong Li, Xinghao Chen, Yehui Tang, and Yunhe Wang. 2025. [EAQuant: Enhancing post-training quantization for MoE models via expert-aware optimization](#). *arXiv preprint arXiv:2506.13329*.
- Intel. 2026. [AutoRound: Advanced quantization algorithm for large language models](#). Software, version v0.9.7.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23*, pages 611–626, New York, NY, USA. Association for Computing Machinery.
- Pingzhi Li, Xiaolong Jin, Yu Cheng, and Tianlong Chen. 2024. [Examining post-training quantization for mixture-of-experts: A benchmark](#). *arXiv preprint arXiv:2406.08155*.
- Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, and 9 others. 2025. [Muon is scalable for LLM training](#). *arXiv preprint arXiv:2502.16982*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- NVIDIA. 2025. [Nemotron-post-training-dataset-v1](#). Hugging Face dataset. CC BY 4.0.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. [OmniQuant: Omnidirectionally calibrated quantization for large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). In *International Conference on Learning Representations*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [MMLU-Pro: A more robust and challenging multi-task language understanding benchmark](#). In *Advances in Neural Information Processing Systems*, volume 37. Datasets and Benchmarks Track.
- 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, and 41 others. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *arXiv preprint arXiv:2311.07911*.

Model	τ	PPL $\downarrow$	Avg. Acc. $\uparrow$
<i>4-bit weight-only quantization</i>			
DeepSeek-V2-Lite	τ^*	8.30	40.52
	1.00	8.56	34.96
Moonlight-16B-A3B	τ^*	9.13	50.44
	1.00	9.41	47.16
Qwen3-30B-A3B	τ^*	9.08	75.24
	1.00	9.24	74.36
<i>3-bit-average weight-only quantization</i>			
DeepSeek-V2-Lite	τ^*	10.72	33.38
	1.00	11.06	29.37
Moonlight-16B-A3B	τ^*	13.93	40.57
	1.00	13.75	39.20
Qwen3-30B-A3B	τ^*	11.29	69.72
	1.00	12.30	61.88

Table 4: Effect of the layer-selection threshold τ on the downstream-router loss $\mathcal{L}_{\text{dr}}$ in MENDS-MoE. τ^* denotes the configuration used in our main experiments, where $\mathcal{L}_{\text{dr}}$ is applied only to the MoE blocks selected by our overlap-based activation criterion; $\tau = 1.00$ applies $\mathcal{L}_{\text{dr}}$ to all eligible MoE blocks with an available next router. We report WikiText-2 perplexity (PPL) and the unweighted mean accuracy (Avg. Acc.) over five downstream benchmarks—MMLU-Pro, GSM8K-CoT, GPQA-Diamond, HumanEval, and IFEval—under 4-bit and 3-bit-average weight-only quantization. Lower PPL and higher Avg. Acc. are better; the better value per row is in **bold**.

A Effect of Applying Downstream Supervision to All Blocks

Our main method activates the downstream router loss only for blocks whose RTN-pilot overlap falls below the threshold τ . To examine whether this adaptive design is necessary, we compare it with a variant that applies $\mathcal{L}_{\text{dr}}$ to every block with an available next MoE router, regardless of its measured overlap.

As shown in Table 4, applying downstream supervision to all eligible blocks generally degrades quantized-model performance relative to our adaptive strategy, reducing average accuracy in all six settings. We hypothesize that the downstream loss is most useful when quantization is likely to change next-router expert membership. For blocks whose next-router decisions are already largely preserved, imposing the same auxiliary loss provides little corrective benefit and can over-constrain the calibration objective. In particular, unnecessary downstream constraints may interfere with block reconstruction or restrict quantization solutions that preserve model outputs well without introducing meaningful routing errors. The overlap-based ac-

Table 5: Hyperparameters used for MENDS-MoE. DRL (%) denotes the percentage of MoE blocks for which the downstream-router loss is activated. The shared settings are $\eta = 0.7$, $b_{\text{sel}} = 1$, and $b_{\text{out}} = 2$.

Model	Bits	λ_1	λ_2	τ	DRL (%)
DeepSeek-V2-Lite	4-bit	0.010	0.0010	0.865	15.4
	3-bit avg.	0.010	0.0010	0.850	34.6
Moonlight-16B-A3B	4-bit	0.010	0.0010	0.925	11.5
	3-bit avg.	0.010	0.0010	0.875	26.9
Qwen3-30B-A3B	4-bit	0.005	0.0005	0.955	14.9
	3-bit avg.	0.005	0.0005	0.940	19.1

tivation rule therefore focuses next-router supervision on blocks for which it provides a useful corrective signal.

B Additional Experimental Details

Hyperparameter configuration. We use a fixed boundary construction throughout all experiments, setting $b_{\text{sel}} = 1$ and $b_{\text{out}} = 2$, together with a margin relaxation factor of $\eta = 0.7$. The coefficients λ_1 and λ_2 control the relative contributions of the current-router and downstream-router losses, respectively. We set these coefficients by inspecting the gradient magnitudes of the reconstruction and routing objectives on the first three blocks so that no auxiliary objective dominates the reconstruction objective. Their selected values are reported in Table 5.

For each model and precision setting, we first compute the RTN-pilot overlap $\sigma_{\ell \rightarrow \ell+1}$ for every MoE block and inspect its distribution. We then shortlist three nearby τ candidates around the central or transition region of the observed overlap range and select the final value using validation perplexity. For Qwen3, whose overlap scores are concentrated in a narrower high-overlap range, we use a finer local candidate spacing. Table 5 reports the selected thresholds and the resulting percentage of MoE blocks for which $\mathcal{L}_{\text{dr}}$ is activated.

Benchmark protocols. We evaluate language-modeling quality by computing WikiText-2 perplexity on non-overlapping sequences of 2,048 tokens, following the GPTQ evaluation protocol (Merity et al., 2017; Frantar et al., 2023). For downstream evaluation, MMLU-Pro is evaluated with five in-context demonstrations, whereas GSM8K-CoT, GPQA-Diamond, HumanEval, and IFEval are evaluated in the zero-shot setting. All generation-based evaluations use greedy decoding. We report `exact_match` for MMLU-Pro, GSM8K-CoT,

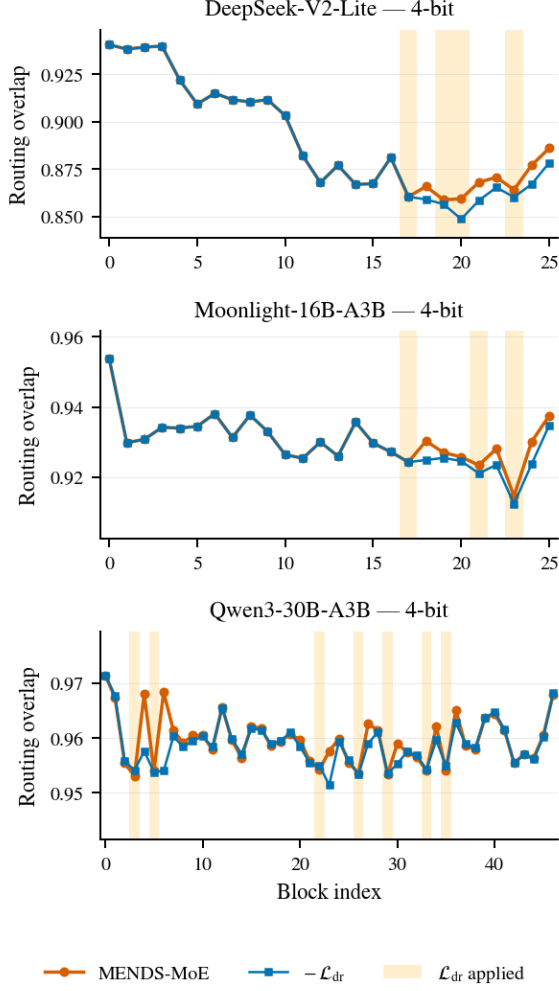


Figure 2: Next-router top- k overlap under 4-bit weight-only quantization, averaged over 65,536 probe tokens. Shaded regions indicate blocks selected for downstream supervision.

and GPQA-Diamond, pass@1 for HumanEval, and prompt_level_strict_acc for IFEval.

Generation limits. We set the maximum generation length to 3,072 tokens for MMLU-Pro, GSM8K-CoT, HumanEval, and IFEval. GPQA-Diamond is assigned a larger budget because its graduate-level science questions can require longer reasoning traces. For DeepSeek-V2-Lite-Chat and Qwen3-30B-A3B, we allow up to 8,192 generated tokens on GPQA-Diamond. For Moonlight-16B-A3B-Instruct, whose context window is limited to 8,192 tokens, we cap the GPQA-Diamond generation length at 4,096 tokens so that both the prompt and generated continuation fit within the available context.

C Routing-Overlap Analysis under 4-bit Quantization

Figure 2 extends the routing-overlap analysis in the main text to the 4-bit weight-only quantization setting. As in the 3-bit-average analysis, MENDS-MoE and its $-\mathcal{L}_{dr}$ ablation are quantized with the same random seed and therefore follow identical overlap trajectories until downstream supervision is first activated. After activation, the full method generally maintains or recovers higher overlap with the full-precision next-router expert sets across activated blocks. Together with the main-text analysis, these results indicate that $\mathcal{L}_{dr}$ can improve next-router selection preservation under both evaluated precision settings.