SRA-MoE: Output-Aware Selective Router Alignment for MoE Quantization

Geonho Lee [1 2 †] **Hancheol Park** [1] **Seunghyun Lee** [3] **Jungwook Choi** [2] **Tae-Ho Kim** [1]

Abstract

Mixture-of-Experts (MoE) architectures enable scalable large language models (LLMs), but their deployment remains memory-intensive, making quantization essential. However, quantizing MoE models introduces routing shifts, where expert selection differs from the baseline model and can degrade performance. Existing router alignment methods uniformly minimize routing discrepancies across all tokens, implicitly treating all routing shifts as equally important. In this work, we show routing shifts exhibit highly heterogeneous impact on model outputs: while some routing shifts substantially affect output behavior, many others induce negligible output discrepancy despite large routing changes. Motivated by this observation, we propose Selective Router Alignment (SRA), an output-aware alignment strategy that prioritizes optimization on tokens exhibiting meaningful output discrepancy after quantization. Experiments across multiple MoE LLMs and reasoning benchmarks show that SRA generally improves over conventional router alignment. Our findings suggest that effective MoE router alignment depends not only on reducing router shifts, but also on prioritizing those that meaningfully affect output behavior.

1. Introduction

Mixture-of-Experts (MoE) architectures (Jacobs et al., 1991; Jordan & Jacobs, 1993; Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024) have become a key paradigm for scaling large language models (LLMs). By activating only a subset of experts for each token, MoE models achieve substantial parameter scaling with relatively low computation cost, leading many recent frontier LLMs to adopt MoE architecture (DeepSeek-AI, 2026; Qwen Team, 2026; Park et al., 2025; LG AI Research, 2025; NVIDIA, 2025a; DeepMind, 2026; GLM-5-Team et al., 2026).

Despite their efficiency, MoE models still incur substantial inference-time memory overhead because all expert parameters reside in memory. Quantization is therefore essential for practical deployment. Recent post-training quantization (PTQ) methods significantly reduce memory usage while preserving model quality (Frantar et al., 2023; Lin et al., 2024b;a; Cheng et al., 2025). However, quantizing MoE models introduces routing shift, where quantization error alters routing decisions. Prior work shows that such routing shift degrades quantized MoE performance and proposes router alignment methods to match baseline routing behavior (Chen et al., 2025a; Fu et al., 2025; Park et al., 2026).

Existing router alignment approaches uniformly minimize routing perturbations across all tokens, implicitly assuming that all routing shifts are equally important. However, we observe that routing perturbations exhibit highly heterogeneous effects on model outputs: while some routing shifts alter output behavior, many others induce negligible output discrepancy despite large routing changes. This distinction is particularly important for autoregressive generation, where observable output deviation can alter subsequent decoding trajectories (Lee et al., 2024; Li et al., 2026). In contrast, tokens whose output distributions remain nearly unchanged after quantization are less likely to affect generation behavior, even when perturbations occur.

Motivated by this observation, we propose Selective Router Alignment (SRA), an output-aware alignment strategy that prioritizes router alignment on tokens exhibiting meaningful output discrepancy after quantization.

Experiments on multiple MoE LLMs show that SRA generally improves over conventional router alignment across diverse reasoning benchmarks, with particularly strong gains under more aggressive low-bit quantization settings.

Our contributions are summarized as follows:

- We analyze routing perturbations in quantized MoE models, showing that their impact on output discrepancy varies widely and that many large routing shifts cause negligible output discrepancy, making uniform

[†]Work done during an internship at Nota Inc. [1]Nota Inc., Seoul, South Korea [2]Hanyang University, Seoul, South Korea [3]Upstage, Seoul, South Korea. Correspondence to: Jungwook Choi <choij@hanyang.ac.kr>, Tae-Ho Kim <thkim@nota.ai>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

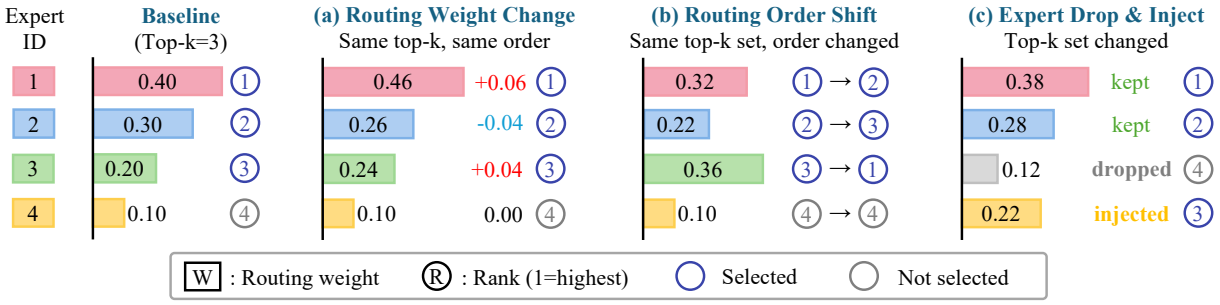


Figure 1. Types of routing perturbation in quantized MoE models. Quantization can alter routing behavior through routing weight changes, routing order shifts, or expert drop/inject events. While routing weight changes preserve both expert selection and relative ranking, routing order shifts reflect sufficiently large perturbations that alter relative expert preference while the selected top-k set. Routing shift refers to structural discrepancies that modify expert rankings or selected expert sets.

alignment ineffective.

- We propose Selective Router Alignment (SRA), which prioritizes router alignment on tokens exhibiting meaningful output discrepancy.
- We demonstrate that SRA generally improves over conventional router alignment across multiple models and quantization settings.

2. Background

2.1. Mixture-of-Experts Architecture

MoE models adopt a sparse activation, where only a subset of experts is selected for each input token. Given an input representation x, a router selects the top-k expert set $\mathcal{E}_k(x)$ and corresponding routing weight $w_i(x)$. Where $E_i(x)$ denotes the output of i-th expert, the final MoE output is computed as:

$$y = \sum_{i \in \mathcal{E}_k(x)} w_i(x) \cdot E_i(x). \tag{1}$$

2.2. Quantization in Large Language Models

Quantization reduces the memory footprint of LLMs by representing parameters using lower numerical precision. In MoE models, weight-only quantization is commonly used to alleviate memory constraints, enabling a 100B-scale MoE model that typically requires at least four 80GB GPUs to run on a single GPU with 4-bit quantization.

Recent PTQ methods improve performance by minimizing reconstruction error (Frantar et al., 2023; Shao et al., 2024; Cheng et al., 2025) or improving quantization-friendly parameterization (Lin et al., 2024b;a), substantially reducing memory usage while preserving model quality.

2.3. MoE-specific Quantization Strategies

Applying quantization to MoE models is challenging due to expert heterogeneity and sparse activation. Since experts differ in importance and activation frequency(Su et al., 2025), prior work explores expert-wise mixed-precision quantization (Li et al., 2024; Huang et al., 2025; Chowdhury et al., 2026) and expert-balanced calibration data (Chen et al., 2025b), highlighting the importance of incorporating MoE-specific routing behavior into quantization design.

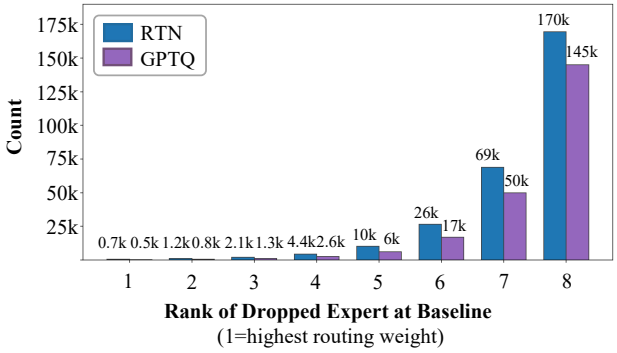


Figure 2. Distribution of dropped expert positions under quantization on Qwen3-30B-A3B using the Pile dataset. Lower-ranked experts are more frequently affected by routing perturbation. Advanced PTQ methods such as GPTQ mitigate these discrepancies, although some routing differences still emerge after quantization.

2.4. Routing Shift in Quantized MoE Models

In quantized MoE models, quantization can perturb routing behavior, causing routing shifts between baseline and quantized models. Although routers are typically kept in full precision because of their small size, errors from preceding layers can still alter routing decisions. As illustrated in Figure 1, quantization can alter routing behavior by changing routing weights while preserving expert selection (a), modifying expert ordering (b), or replacing selected experts (c); we focus on structural shifts affecting ranking or selection.

Figure 2 shows that expert drops occur more frequently among lower-ranked experts. Although advanced PTQ methods such as GPTQ (Frantar et al., 2023) mitigate routing perturbations, routing perturbations remain. These observations motivate further study of routing perturbations in quantized MoE models.

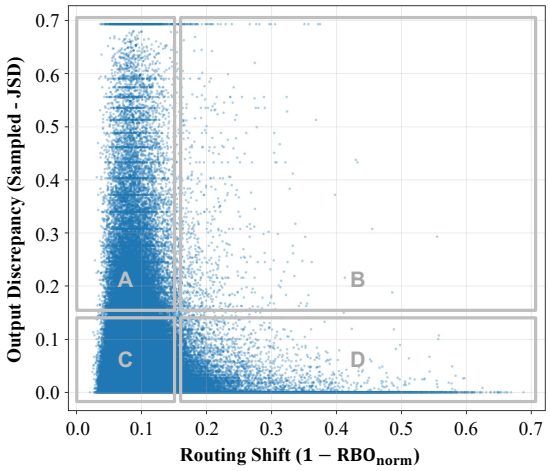


Figure 3. Relationship between routing shift and output discrepancy on Qwen3-30B-A3B under W4A16. Each point represents a token. Many tokens retain similar output distributions despite large routing shifts, indicating generation behavior may remain stable after quantization. Similar heterogeneous relationships are consistently observed across datasets and bit-widths (Appendix B).

2.5. Router Alignment

To mitigate routing shift, prior work proposes router alignment methods that encourage the quantized model to mimic the routing behavior of the baseline model, showing that preserving routing consistency improves quantized MoE performance (Chen et al., 2025a). Existing methods minimize routing perturbations across all tokens using value-, distribution-, and structural-alignment objectives. (Chen et al., 2025a; Fu et al., 2025; Fang & Huang, 2025; Park et al., 2026). In contrast, our work focuses on determining which routing perturbations should be prioritized during alignment.

3. Analysis

In this section, we analyze the relationship between routing shift and output discrepancy in quantized MoE models. We first introduce metrics for both quantities and then present empirical findings showing that routing shifts differ substantially in their impact on output behavior.

3.1. Output Discrepancy Metric: Sampled-JSD

To measure output discrepancy between baseline and quantized models, we use JSD over token-level probability distributions. Since generation is mainly determined by a small set of candidate tokens after sampling filters such as top-k or top-p truncation, we propose Sampled-JSD, which computes JSD over the union of candidate token sets considered by both models:

$$\mathrm{JSD}_{\mathrm{Sampled}} = \mathrm{JSD}(p_U^*, \hat{p}_U), \tag{2}$$

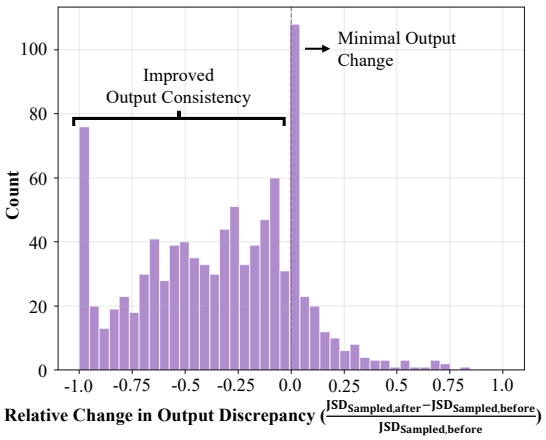


Figure 4. Effect of replacing quantized routing decisions with baseline routing for tokens in Region B of Figure 3. Negative values indicate reduced output discrepancy after routing replacing. Correcting routing perturbation frequently restores output behavior when observable output discrepancy exists after quantization.

where p_U^* and $\hat{p}_U$ denote the distributions restricted to the union candidate set U. By focusing on competing candidates, Sampled-JSD reduces the influence of low-probability tokens and better reflects generation-time output discrepancy. Appendix A provides additional implementation details and comparisons with standard full-vocabulary JSD.

3.2. Routing Shift Metric: RBO-based Score

To quantify routing shift, we compare the top-k expert rankings of the baseline and quantized models. As shown in Figure 1, routing perturbations may alter expert orderings or the selected expert set. To capture these structural routing shifts, we use Rank-Biased Overlap (RBO) (Webber et al., 2010), which emphasizes higher-ranked positions:

$$\mathrm{RBO}(S, T, p) = (1-p)\sum_{d=1}^{k} p^{d-1} \cdot A_d. \tag{3}$$

Where A_d denotes the overlap between the top-d prefixed of the two ranked lists S and T,

$$A_d = \frac{|\{s_1, \ldots, s_d\} \cap \{t_1, \ldots, t_d\}|}{d}. \tag{4}$$

The weighting factor $p \in (0, 1)$ ($p = 0.9$ in our experiments) emphasizes agreement at higher-ranked positions, making RBO suitable for comparing top-k routing behavior in MoE models. We then define the routing shift as:

$$\text{Routing Shift} = 1 - \mathrm{RBO}_{\mathrm{norm}}, \tag{5}$$

where RBO_{norm} denotes RBO normalized to the range [0,1]. Routing shift ranges from 0 (identical routing) to 1 (completely different expert selections), capturing perturbations in both expert selection and ordering.

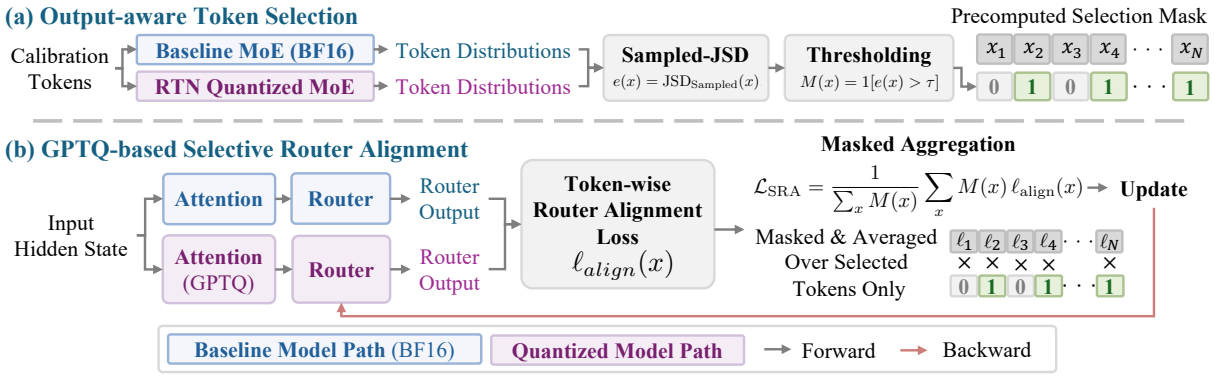


Figure 5. Overview of Selective Router Alignment (SRA). SRA first identifies tokens exhibiting observable output discrepancy after quantization and selectively applies router alignment only to those tokens. Tokens whose output distributions remain nearly unchanged are excluded from optimization.

3.3. Routing Shifts Exhibit Heterogeneous Impact

We perform inference using both baseline and quantized MoE models and measure output discrepancy and routing shift for each token. As shown in Figure 3, many tokens preserve highly similar output distributions after quantization despite substantial routing shifts, indicating that numerous routing perturbations remain effectively output-preserving.

This observation is important because autoregressive generation is highly sensitive to output deviations (Lee et al., 2024; Li et al., 2026). Once generation diverges, subsequent decoding trajectories may increasingly drift from the baseline model. These findings suggest that router alignment should prioritize tokens with meaningful output discrepancies, rather than uniformly optimizing all tokens.

To better characterize this behavior, we partition the scatter plot into four regions. Region A contains tokens with low routing shift but high output discrepancy, indicating that non-routing quantization errors can independently affect generation. Region B contains tokens with both high routing shift and high output discrepancy, where routing perturbations coincide with observable output degradation. Region C contains the majority of tokens where both remain low, while Region D contains tokens with large routing shift but near-zero output discrepancy, indicating that many routing shifts do not meaningfully affect outputs.

To isolate the effect of routing perturbations, we replace Region B routing decisions in the quantized model with those from the baseline model while keeping all other components unchanged. As shown in Figure 4, correcting routing shifts is often associated with reduced output discrepancy, suggesting that router alignment can help recover generation behavior when quantization causes observable output deviations. Additional analysis is provided in Appendix C.

Overall, these results suggest that router alignment remains important, but not all routing shifts should be treated equally. Uniformly aligning all routing perturbations may dilute optimization toward tokens whose outputs are already preserved after quantization. Instead, router alignment should prioritize routing shifts that manifest as meaningful output discrepancy.

4. Method: Selective Router Alignment

Motivated by the heterogeneous impact of routing shifts on model outputs, we propose Selective Router Alignment (SRA), an output-aware router alignment strategy that focuses on tokens exhibiting meaningful output discrepancy after quantization. As illustrated in Figure 5, SRA consists of (i) target token selection based on output discrepancy and (ii) selective router alignment on the selected tokens.

4.1. Output-aware Token Selection

To estimate token-wise output discrepancy, we compute Sampled-JSD between the output distributions of the baseline and RTN-quantized models. RTN efficiently approximates quantization-induced output discrepancy without iterative PTQ optimization. For a calibration token x, token-wise output discrepancy is defined as:

$$e(x) = \mathrm{JSD}_{\mathrm{Sampled}}(x). \tag{6}$$

We construct a binary selection mask using the discrepancy:

$$M(x) = 1[e(x) > \tau], \tag{7}$$

where τ denotes the discrepancy threshold and $M(x) = 1$ indicates that token x is selected for alignment optimization. This selection mask is precomputed prior to router alignment and reused throughout PTQ optimization.

In practice, we use a small threshold τ to exclude tokens with near-zero output discrepancy after quantization. As shown in Figure 3 and Appendix B, a substantial fraction of tokens exhibit near-zero output discrepancy despite routing perturbations, indicating preserved generation behavior.

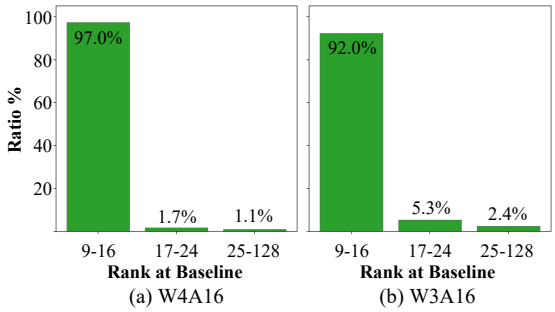


Figure 6. Distribution of injected expert positions on Qwen3-30B-A3B (top-k=8) using the C4 dataset. Most injected experts originate from ranks near the original top-k experts, while substantially lower-ranked experts are comparatively rare after quantization.

Empirically, reducing τ below 0.0001 changes the selected token ratio only marginally, indicating most excluded tokens already lie near the zero-discrepancy regime.

4.2. Selective Router Alignment

We integrate SRA into GPTQ (Frantar et al., 2023)-based router alignment by selectively masking token contributions during alignment optimization. GPTQ performs layer-wise quantization by sequentially optimizing quantized layers and propagating intermediate hidden states through the quantized model. During this process, router alignment minimizes discrepancies between the router outputs of the baseline and quantized model paths.

Given a calibration token x, we define a token-wise router alignment loss $\ell_{\text{align}}(x)$ using router outputs of baseline and quantized model paths. Unlike conventional router alignment, which aggregates losses over all calibration tokens, SRA applies the mask during loss aggregation:

$$\mathcal{L}_{\text{SRA}} = \frac{1}{\sum_x M(x)} \sum_x M(x)\, \ell_{\text{align}}(x) \tag{8}$$

Thus, all calibration tokens participate in router forward computation, while only selected tokens contribute to the alignment objective. This preserves the original router alignment procedure while prioritizing optimization toward selected tokens with larger output discrepancy.

4.3. Top-3K MSE

Prior router alignment methods have observed that aligning the full routing distribution can introduce noise from low-relevance experts with negligible routing weights (Chen et al., 2025a), proposing restricted alignment to an expanded subset of high-ranked experts beyond the original top-k.

To analyze how far routing perturbations extend beyond the original routing boundary, we measure baseline ranks of experts newly selected after quantization. As shown in Figure 6, most injected experts originate within the baseline top-$3k$, while much lower-ranked experts are rarely selected.

These observations suggest that routing perturbations are largely localized near the routing boundary, and that considering substantially lower-ranked experts may primarily introduce noise into the alignment objective. Motivated by this observation, we compute the router alignment loss over the baseline top-$3k$ experts:

Let $\mathcal{E}_{3k}(x)$ denote the top-$3k$ experts for token x. The alignment loss is:

$$\ell_{align}(x) = \frac{1}{|\mathcal{E}_{3k}(x)|} \sum_{i \in \mathcal{E}_{3k}(x)} \left(r_{b,i}(x) - r_{q,i}(x)\right)^2, \tag{9}$$

where $r_{b,i}(x)$ and $r_{q,i}(x)$ denote the router outputs from the baseline and quantized model paths, respectively.

Using Top-3K MSE allows the alignment objective to capture most meaningful injected experts while reducing the influence of substantially lower-ranked experts with negligible routing weights.

Table 1. Main results on Qwen3-30B-A3B under W4A16 and W3A16 quantization. SRA generally improves over conventional router alignment while applying alignment only to tokens exhibiting observable output discrepancy.

Bit	Align	AIME24	AIME25	GPQA-D	MMLU-P
Baseline		80.42	78.75	61.30	72.81
GPTQ W4A16	-	79.58	67.50	60.54	71.09
	RA	79.58	67.50	60.73	71.09
	SRA	79.58	**70.00**	**61.87**	**71.10**
GPTQ W3A16	-	66.67	48.75	53.66	64.47
	RA	69.58	52.50	**55.50**	64.73
	SRA	**72.08**	**55.42**	54.05	**67.17**

5. Experiments

5.1. Experimental Setup

Models and Benchmarks. We evaluate SRA on recent MoE LLMs, including Qwen3-30B-A3B (Qwen Team, 2025), Solar-Open-100B (Park et al., 2025), and GLM-4.5-Air (GLM Team et al., 2025), using AIME24, AIME25 (Maxwell-Jia, 2025), GPQA-Diamond(GPQA-D) (Rein et al., 2024), and MMLU-Pro(MMLU-P) (Wang et al., 2024). AIME uses pass@4, while remaining benchmarks use pass@1. AIME and GPQA-D are evaluated with 16k generation length and averaged over eight runs, while MMLU-P uses 4k generation length with a single run.

Quantization and Router Alignment. We adopt GPTQ for PTQ under W4A16 and W3A16 settings, while keeping embedding layer, LM head, and router in full precision. We

Table 2. Results on large-scale MoE models under W4A16 quantization. Conventional router alignment shows varying effectiveness across models and benchmarks, while SRA achieves competitive or superior quantized performance in most settings.

Model	Method	AIME24	AIME25	GPQA-D	MMLU-P
Solar Open 100B	Baseline	77.92	62.50	58.59	65.92
	GPTQ	73.33	60.00	**59.60**	63.54
	RA	70.00	61.25	58.33	63.80
	SRA	**76.67**	**63.75**	57.01	**65.79**
GLM 4.5 Air	Baseline	72.08	51.67	69.63	64.5
	GPTQ	64.58	51.25	68.18	61.69
	RA	63.75	**52.08**	67.87	62.08
	SRA	**67.92**	51.67	**69.26**	**62.96**

Table 3. Ablation study on alignment loss under W3A16 quantization. Top-3K MSE achieves the strongest performance on MMLU-Pro by balancing coverage of routing shifts and robustness to low-relevance experts.

Loss Type	Top-K	Top-2K	Top-3K	Full
MMLU-Pro	63.93	63.92	**64.73**	64.23

compare: (i) GPTQ without alignment, (ii) conventional router alignment (RA), (iii) the proposed selective router alignment (SRA). Calibration data is sampled from the Pile dataset (Gao et al., 2020).

5.2. Main Results on Qwen3-30B-A3B

Table 1 summarizes improved performance over GPTQ without alignment. SRA generally improves over conventional RA on most benchmarks, with particularly noticeable gains under the more aggressive W3A16 setting. Under W4A16, SRA achieves competitive or slightly improved performance across evaluation tasks.

With $\tau = 0.0001$, about 49% (W4A16) and 57% (W3A16) of tokens are selected, indicating that many tokens preserve output distributions after quantization despite routing shifts.

Despite excluding these near-zero discrepancy tokens from alignment, SRA generally maintains or improves performance relative to uniform router alignment, suggesting that prioritizing tokens with observable output discrepancy can provide more effective alignment signals.

5.3. Scaling to Larger MoE Models

We further evaluate SRA on Solar-Open-100B (Park et al., 2025) and GLM-4.5-Air (GLM Team et al., 2025) under W4A16. Results are summarized in Table 2. Conventional RA generally improves over GPTQ, although its effectiveness varies across models and benchmarks. Across most evaluation settings, SRA achieves competitive or improved quantized performance relative to conventional RA.

Table 4. Ablation study on aggressive token selection under W3A16 quantization. Restricting alignment to tokens exhibiting both high routing shift and high output discrepancy improves some benchmarks but reduces stability across tasks compared to the default SRA configuration.

τ	R_{th}	AIME24	AIME25	GPQA-D	MMLU-P
0	0	69.58	52.50	**55.50**	64.73
0.1	0.1	**80.42**	51.28	53.41	64.71
0.0001	0	72.08	**55.42**	54.05	**67.17**

5.4. Ablation Study

Loss Type. We compare Top-K, Top-2K, Top-3K, and Full MSE losses on Qwen3-30B-A3B under W3A16. Results are summarized in Table 3. Among the evaluated configurations, Top-3k achieves the strongest performance on MMLU-Pro. These observations suggest that Top-3k MSE provides a reasonable balance between coverage and noise.

Aggressive Token Selection. We additionally study a more restrictive selection strategy that aligns only tokens exhibiting both high routing shift and high output discrepancy. As shown in Table 4, although this approach can substantially improve certain benchmarks, for example, achieving near-baseline performance on AIME24 under W3A16, it introduces less stable results across tasks.

In contrast, the default SRA configuration, which selects tokens solely based on output discrepancy using a mild threshold, achieves more consistent overall performance. These results suggest that overly restrictive token selection may reduce performance stability across evaluation tasks.

Conclusion

In this work, we revisit router alignment in quantized MoE models from an output-centric perspective. Our analysis shows that many tokens preserve nearly identical output distributions after quantization, even under substantial routing perturbations. Since autoregressive generation is particularly sensitive to observable output deviations, uniformly aligning tokens whose outputs already remain unchanged may provide limited optimization benefit. Motivated by this observation, we propose Selective Router Alignment (SRA), which selectively applies router alignment only to tokens exhibiting meaningful output discrepancy after quantization. Experiments across multiple MoE LLMs demonstrate that SRA generally improves over conventional router alignment, particularly under aggressive low-bit quantization settings. These findings suggest that prioritizing output-relevant routing discrepancies can provide a useful optimization strategy for quantized MoE alignment.

Acknowledgements

This research was conducted as part of the Sovereign AI Foundation Model Project (GPU Track), organized by the Ministry of Science and ICT (MSIT) and supported by the National IT Industry Promotion Agency (NIPA), South Korea (PJT-26-010017).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Chen, Y., Shao, Y., Wang, P., and Cheng, J. 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)*, pp. 12942–12963, 2025a.

Chen, Z., Hu, X., Yang, D., Xu, Z., Chen, X., Yuan, Z., Zhou, S., and Yu, J. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. In *International Conference on Machine Learning*, pp. 8245–8260. PMLR, 2025b.

Cheng, W., Zhang, W., Guo, H., and Shen, H. Signroundv2: Closing the performance gap in extremely low-bit post-training quantization for llms. *arXiv preprint arXiv:2512.04746*, 2025.

Chowdhury, M. N. R., El Maghraoui, K., Tsai, H., Wang, N., Burr, G. W., Liu, L., and Wang, M. Efficient quantization of mixture-of-experts with theoretical generalization guarantees. In *The Fourteenth International Conference on Learning Representations*, 2026.

Dai, D., Deng, C., Zhao, C., Xu, R., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024.

DeepMind, G. Gemma 4 model card, 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4. Accessed: 2026-05-04.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., and Gardner, M. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 1286–1305, 2021.

Du, W., Toshniwal, S., Kisacanin, B., Mahdavi, S., Moshkov, I., Armstrong, G., Ge, S., Minasyan, E., Chen, F., and Gitman, I. Nemotron-math: Efficient long-context distillation of mathematical reasoning from multi-mode supervision. *arXiv preprint arXiv:2512.15489*, 2025.

Fang, Y.-Z. and Huang, J.-D. Router choice matters: Rank-aware post-training quantization for moe models, 2025. URL https://openreview.net/forum?id=kPgLp47bJf.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*. OpenReview, 2023.

Fu, Z., Zhao, T., Ding, N., Yu, X., Li, X., Tang, Y., and Wang, Y. Eaquant: Enhancing post-training quantization for moe models via expert-aware optimization. *arXiv preprint arXiv:2506.13329*, 2025.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

GLM-5-Team, :, Zeng, A., Lv, X., Hou, Z., Du, Z., Zheng, Q., Chen, B., Yin, D., Ge, C., Huang, C., Xie, C., Zhu, C., Yin, C., Wang, C., Pan, G., Zeng, H., Zhang, H., Wang, H., Chen, H., Zhang, J., Jiao, J., Guo, J., Wang, J., Du, J., Wu, J., Wang, K., Li, L., Fan, L., Zhong, L., Liu, M., Zhao, M., Du, P., Dong, Q., Lu, R., Shuang-Li, Cao, S., Liu, S., Jiang, T., Chen, X., Zhang, X., Huang, X., Dong, X., Xu, Y., Wei, Y., An, Y., Niu, Y., Zhu, Y., Wen, Y., Cen, Y., Bai, Y., Qiao, Z., Wang, Z., Wang, Z., Zhu, Z., Liu, Z., Li, Z., Wang, B., Wen, B., Huang, C., Cai, C., Yu, C., Li, C., Hu, C., Zhang, C., Zhang, D., Lin, D., Yang, D., Wang, D., Ai, D., Zhu, E., Yi, F., Chen, F., Wen, G., Sun, H., Zhao, H., Hu, H., Zhang, H., Liu, H., Zhang, H., Peng, H., Tai, H., Zhang, H., Liu, H., Wang, H., Yan, H., Ge, H., Liu, H., Chu, H., Zhao, J., Wang, J., Zhao, J., Ren, J., Wang, J., Zhang, J., Gui, J., Zhao, J., Li, J., An, J., Li, J., Yuan, J., Du, J., Liu, J., Zhi, J., Duan, J., Zhou, K., Wei, K., Wang, K., Luo, K., Zhang, L., Sha, L., Xu, L.,

Wu, L., Ding, L., Chen, L., Li, M., Lin, N., Ta, P., Zou, Q., Song, R., Yang, R., Tu, S., Yang, S., Wu, S., Zhang, S., Li, S., Li, S., Fan, S., Qin, W., Tian, W., Zhang, W., Yu, W., Liang, W., Kuang, X., Cheng, X., Li, X., Yan, X., Hu, X., Ling, X., Fan, X., Xia, X., Zhang, X., Zhang, X., Pan, X., Zou, X., Zhang, X., Liu, Y., Wu, Y., Li, Y., Wang, Y., Zhu, Y., Tan, Y., Zhou, Y., Pan, Y., Zhang, Y., Su, Y., Geng, Y., Yan, Y., Tan, Y., Bi, Y., Shen, Y., Yang, Y., Li, Y., Liu, Y., Wang, Y., Li, Y., Wu, Y., Zhang, Y., Duan, Y., Zhang, Y., Liu, Z., Jiang, Z., Yan, Z., Zhang, Z., Wei, Z., Chen, Z., Feng, Z., Yao, Z., Chai, Z., Wang, Z., Zhang, Z., Xu, B., Huang, M., Wang, H., Li, J., Dong, Y., and Tang, J. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

GLM Team, Zeng, A., Lv, X., Zheng, Q., Hou, Z., Chen, B., Xie, C., Wang, C., Yin, D., Zeng, H., Zhang, J., Wang, K., Zhong, L., Liu, M., Lu, R., Cao, S., Zhang, X., Huang, X., Wei, Y., Cheng, Y., An, Y., Niu, Y., Wen, Y., Bai, Y., Du, Z., Wang, Z., Zhu, Z., Zhang, B., Wen, B., Wu, B., Xu, B., Huang, C., Zhao, C., Cai, C., Yu, C., Li, C., Ge, C., Huang, C., Zhang, C., Xu, C., Zhu, C., Li, C., Yin, C., Lin, D., Yang, D., Jiang, D., Ai, D., Zhu, E., Wang, F., Pan, G., Wang, G., Sun, H., Li, H., Li, H., Hu, H., Zhang, H., Peng, H., Tai, H., Zhang, H., Wang, H., Yang, H., Liu, H., Zhao, H., Liu, H., Yan, H., Liu, H., Chen, H., Li, J., Zhao, J., Ren, J., Jiao, J., Zhao, J., Yan, J., Wang, J., Gui, J., Zhao, J., Liu, J., Li, J., Li, J., Lu, J., Wang, J., Yuan, J., Li, J., Du, J., Du, J., Liu, J., Zhi, J., Gao, J., Wang, K., Yang, L., Xu, L., Fan, L., Wu, L., Ding, L., Wang, L., Zhang, M., Li, M., Xu, M., Zhao, M., Zhai, M., Du, P., Dong, Q., Lei, S., Tu, S., Yang, S., Lu, S., Li, S., Li, S., Shuang-Li, Yang, S., Yi, S., Yu, T., Tian, W., Wang, W., Yu, W., Tam, W. L., Liang, W., Liu, W., Wang, X., Jia, X., Gu, X., Ling, X., Wang, X., Fan, X., Pan, X., Zhang, X., Zhang, X., Fu, X., Zhang, X., Xu, Y., Wu, Y., Lu, Y., Wang, Y., Zhou, Y., Pan, Y., Zhang, Y., Wang, Y., Li, Y., Su, Y., Geng, Y., Zhu, Y., Yang, Y., Li, Y., Wu, Y., Li, Y., Liu, Y., Wang, Y., Li, Y., Zhang, Y., Liu, Z., Yang, Z., Zhou, Z., Qiao, Z., Feng, Z., Liu, Z., Zhang, Z., Wang, Z., Yao, Z., Wang, Z., Liu, Z., Chai, Z., Li, Z., Zhao, Z., Chen, W., Zhai, J., Xu, B., Huang, M., Wang, H., Li, J., Dong, Y., and Tang, J. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

Huang, W., Liao, Y., Liu, J., He, R., Tan, H., Zhang, S., Li, H., Liu, S., and QI, X. Mixture compressor for mixture-of-experts llms gains more. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991. URL https://api.semanticscholar.org/CorpusID:572361.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Jordan, M. and Jacobs, R. Hierarchical mixtures of experts and the em algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pp. 1339–1344 vol.2, 1993. doi: 10.1109/IJCNN.1993.716791.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Lee, J., Park, S., Hong, S., Kim, M., Chang, D.-S., and Choi, J. Improving conversational abilities of quantized large language models via direct preference alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11346–11364, 2024.

LG AI Research. K-exaone technical report. *arXiv preprint arXiv:2601.01739*, 2025.

Li, P., Jin, X., Tan, Z., Cheng, Y., and Chen, T. Quantmoe-bench: Examining post-training quantization for mixture-of-experts. *arXiv preprint arXiv:2406.08155*, 2024.

Li, Z., Li, J., Jiang, G., Song, L., Lian, D., and Wei, Y. Scaling reasoning hop exposes weaknesses: Demystifying and improving hop generalization in large language models. *arXiv preprint arXiv:2601.21214*, 2026.

Lin, H., Xu, H., Wu, Y., Cui, J., Zhang, Y., Mou, L., Song, L., Sun, Z., and Wei, Y. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37: 87766–87800, 2024a.

Lin, J. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100, 2024b.

Maxwell-Jia. Aime 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024, 2025.

NVIDIA. Nvidia nemotron 3: Efficient and open intelligence, 2025a. URL https://arxiv.org/abs/2512.20856. White Paper.

NVIDIA. Nemotron-sft-competitive-programming-v2. https://huggingface.co/datasets/nvidia/Nemotron-SFT-Competitive-Programming-v2, 2025b. Hugging Face dataset.

Park, H., Piao, T., and Kim, T.-H. Notamoequant, 2026. URL https://www.nota.ai/community/notamoequantization-an-moe-specific-quantization-method-for-solar-open-100b. 2026-04-01.

Park, S., Kim, S., Cho, J., Gim, G., Jung, D., Cha, M., Choo, E., Hong, T., Jeong, M., Joo, S., Khang, M., Kim, E., Kim, M., Kim, S., Kim, Y., Lee, H., Lee, S., Lee, S., Park, S., Shin, G., Song, I., Song, W., Yang, S., Yi, S., Yoon, S., Ko, J., Song, S., Choi, K., Lee, H., Kim, S., Chang, D.-S., Cho, K., Choe, J., Lee, H., Lee, J.-G., Lim, K., and Oh, A. Solar open technical report. *arXiv preprint arXiv:2601.07022*, 2025. URL https://huggingface.co/papers/2601.07022.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

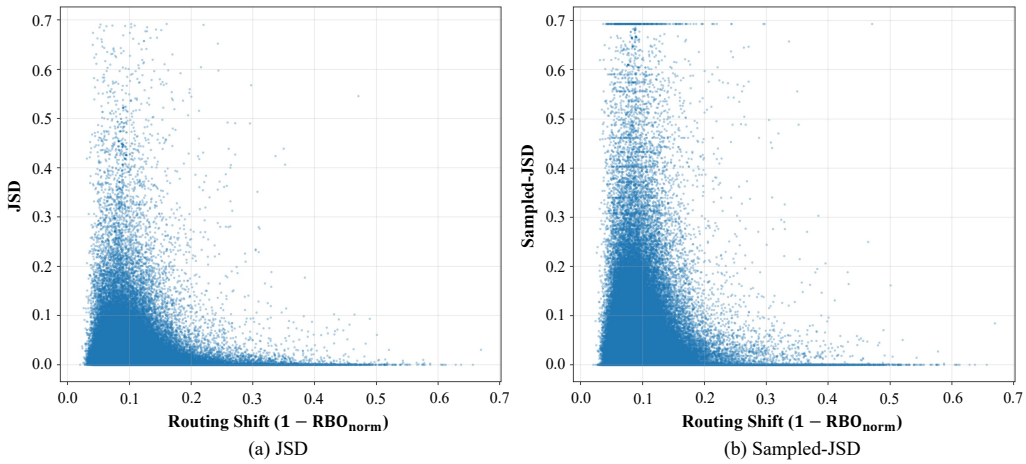
Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Gao, P., Qiao, Y., and Luo, P. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Su, Z., Li, Q., Zhang, H., Ye, W., Xue, Q., Qian, Y., Xie, Y., Wong, N., and Yuan, K. Unveiling super experts in mixture-of-experts large language models. *arXiv preprint arXiv:2507.23279*, 2025.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Webber, W., Moffat, A., and Zobel, J. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), November 2010. ISSN 1046-8188. doi: 10.1145/1852102.1852106. URL https://doi.org/10.1145/1852102.1852106.

A. Additional Details of Sampled-JSD



(a) JSD (b) Sampled-JSD

Figure 7. Comparison between full-vocabulary JSD and Sampled-JSD. Token-wise scatter plots of routing shift against output error measured using either JSD or Sampled-JSD on Qwen3-30B-A3B with the Pile dataset. Sampled-JSD produces larger divergence values and yields a higher concentration of tokens near the upper bound.

To quantify output discrepancy between the baseline and quantized models, we use Jensen-Shannon divergence (JSD) (Lin, 1991) over token-level probability distributions. Let p^* and $\hat{p}$ denote the probability distributions obtained from the baseline and quantized models, respectively. Standard JSD, a symmetric and bounded metric, is defined as:

$$\mathrm{JSD}(p^*, \hat{p}) = \frac{1}{2} D_{\mathrm{KL}}(p^* \parallel m) + \frac{1}{2} D_{\mathrm{KL}}(\hat{p} \parallel m), \tag{10}$$

where $m = \frac{1}{2}(p^* + \hat{p})$.

However, full-vocabulary JSD is not always well aligned with practical decoding behavior, where only a subset of candidate tokens is considered during generation. To better capture generation-time discrepancy, we compute JSD only over the union of candidate token sets considered during sampling. Specifically, we apply the same sampling procedure to both models, including temperature scaling and top-k/top-p filtering, to obtain candidate tokens sets P^* and $\hat{P}$. We then construct the union support:

$$U = P^* \cup \hat{P}. \tag{11}$$

We denote by p^*_U and $\hat{p}_U$ the distributions restricted to the union candidate set U, where tokens not present in the original candidate set are assigned zero probability. The proposed Sampled-JSD is computed over the restricted distributions:

$$\mathrm{JSD}_{\mathrm{Sampled}} = \mathrm{JSD}(p^*_U, \hat{p}_U). \tag{12}$$

By restricting comparison to tokens considered during generation, Sampled-JSD suppresses noise from low-probability regions and better reflects observable output discrepancy under sampling.

The Sampled-JSD is bounded between 0 and $ln2$ (approximately 0.69). A value of 0 indicates identical distributions, while values approaching $ln2$ indicate that the two models place their probability mass on different candidate tokens within the candidate set.

To analyze the behavior of Sampled-JSD, we compare it with standard full-vocabulary JSD. Following the setup in Figure 3, Figure 7 plots token-wise routing shift against output error computed using either JSD or Sampled-JSD.

Both metrics exhibit similar global trends: routing perturbations exhibit highly heterogeneous impact on model outputs. However, Sampled-JSD generally produces sharper separation between negligible and meaningful output discrepancy because it emphasizes differences within the active candidate token set. In contrast, full-vocabulary JSD distributes probability mass across the entire vocabulary, which can dilute meaningful differences with low-probability noise.

These observations suggest that Sampled-JSD provides a more faithful measure of output discrepancy under practical decoding settings.

B. Additional Analysis of Routing Shift and Output Discrepancy

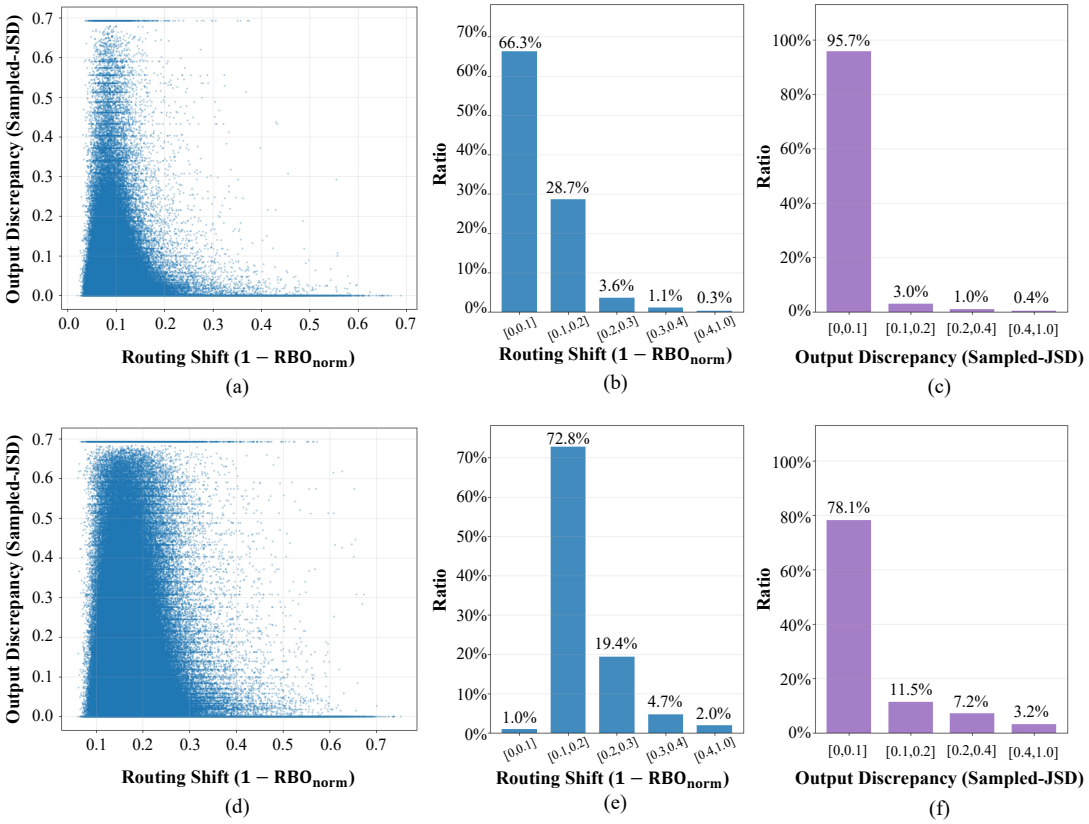


Figure 8. Distribution of routing shift and output discrepancy under different quantization settings. (a,d): Token-wise scatter plots of routing shift and output error. (b,e): Distribution of routing shift across value ranges. (c,f): Distribution of Sampled-JSD across value ranges. (a)-(c) correspond to W4A16, and (d)-(f) corresponds to W3A16 on Qwen3-30B-A3B (using Pile dataset).

Distribution of Routing Shift and Output Error. To complement the scatter plots in Section 3.3, we further analyze the distributions of routing shift and output error using histogram-based binning. Specifically, we report the proportion of tokens falling into different value ranges for both routing shift and Sampled-JSD under 4-bit (W4A16) and 3-bit (W3A16) quantization. The results are shown in Figure 8.

Under the 4-bit setting, both routing shift and output error are heavily concentrated in the low-value region. In particular, more than 95% of tokens exhibit Sampled-JSD values below 0.1, indicating that the majority of tokens preserve highly similar output distributions after quantization. A similar trend is observed for routing shift, where most tokens fall into the lowest range.

In contrast, under the 3-bit setting, we observe a noticeable shift toward higher error values. The proportion of tokens with Sampled-JSD below 0.1 decreases to approximately 78% reflecting increased output discrepancy under more aggressive quantization. Interestingly, routing shift exhibits a different pattern: only a small fraction of tokens (around 1%) fall below 0.1, while the majority are distributed in slightly higher ranges. This suggests that stronger quantization leads to more widespread routing perturbations across tokens.

These results further support our observation that routing shift and output error exhibit different distributional behaviors, and that increasing quantization strength affects them in distinct ways.

Generalization Across Datasets. We further examine whether the observed relationship between routing shift and output error generalizes across different datasets. In addition to the Pile used in the main experiments, we evaluate on datasets with different characteristics, including C4 (Dodge et al., 2021) as another pretraining corpus, and supervised fine-tuning (SFT) datasets such as Nemotron-Math (Du et al., 2025) and Nemotron-Code (NVIDIA, 2025b).

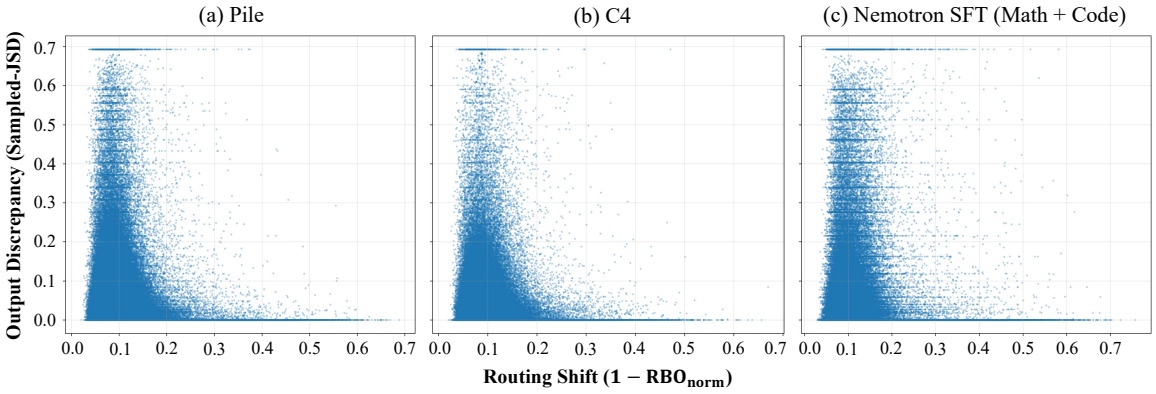


Figure 9. Routing shift and output error across different datasets. Token-wise scatter plots on Qwen3-30B-A3B using the Pile, C4, and Nemotron-SFT datasets, showing similar patterns.

Figure 9 presents token-wise scatter plots of routing shift and output error across three datasets. Overall, we observe consistent patterns across all datasets. These results suggest that our observations are not specific to a particular dataset, but rather reflect a general behavior of quantized MoE models under different data distributions.

C. Additional Analysis of Figure 4

Interestingly, we also observe a small number of cases where injecting baseline routing increases the output error. One possible explanation is that the experts selected by the base-line model may themselves suffer from large quantization error. Another possibility is that, even with identical routing decisions, discrepancies in the input representations to the experts, caused by accumulated quantization noise, lead to mismatches in expert behavior. We leave a deeper investigation of these cases for future work.

D. Inference Efficiency under Quantization

Table 5. Inference efficiency comparison between BF16 and W4A16. Inference is measured on Qwen3-30B-A3B using vLLM with a single H100 80GB GPU, input/output length of 15k tokens, and maximum concurrency of 4. Quantization improves throughput, TTFT, TPOT, and available KV cache capacity compared to BF16.

Bit	Output TPS (tok/s)↑	Total TPS (tok/s)↑	TTFT (ms)↓	TPOT (ms)↓	Available KV Cache (tok)↑
BF16	389.04	778.09	1353.27	10.19	126,368
W4A16	**469.26**	**938.53**	**1333.63**	**8.44**	**583,216**

Although inference acceleration is primarily a benefit of quantization itself rather than the proposed alignment method, we provide additional measurements to illustrate the practical efficiency gains enabled by low-bit MoE deployment.

We measure inference performance on Qwen3-30B-A3B using vLLM (Kwon et al., 2023) with a single H100 80GB GPU. The evaluation is conducted with an input length of 15k tokens, an output length of 15k tokens, and a maximum concurrency of 4 requests. We compare BF16 and W4A16 settings in terms of throughput, time-to-first-token (TTFT), time-per-output-token (TPOT), and available KV cache capacity.

The results are summarized in Table 5. Quantization improves all measured metrics compared to BF16 inference. In particular, TPOT is substantially reduced under W4A16, leading to noticeable improvement in overall throughput. In addition, the reduced model memory footprint enables significantly larger available KV cache memory, allowing more tokens to be cached during inference. This is particularly beneficial for long-context generation scenarios, where KV cache capacity often becomes a major memory bottleneck.