SHADOWMOE: LEARNING EXPERT VULNERABILITY FROM MOE ROUTING TOPOLOGY

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse mixture-of-experts (MoE) models expose a structured trace of computation through token-level routing: experts are not used independently, but repeatedly cooperate across layers. We ask whether this routing topology contains information about expert vulnerability to model compression. We introduce *ShadowMoE*, a topology-aware representation framework that converts token-level top-k routing traces into weighted cross-layer expert graphs and learns relational representations of individual experts. A graph-restricted encoder propagates information only along observed routing paths, while leakage-controlled structural objectives test whether the learned representation captures non-local routing information. We further probe whether these representations predict expert-level quantization residuals and use them to construct budget-constrained mixed-precision policies. On Qwen3-30B-A3B and Mixtual V0.1, ShadowMoE achieves a strict link-prediction AUC of 0.9432, compared with 0.8378 for a graph-free MLP, and its expert representations explain $R^2 = 0.5405$ and 0.5953 of FP8 and INT4 local quantization residuals, respectively. At an average 5.0-bit budget, the resulting INT4/FP8 allocation retains 99.43% and 98.33% of FP8+activation-quantized accuracy on the two models, respectively. Importantly, independently estimated expert risks do not reliably compose into scheme-level degradation: the correlation between predicted local risk and complete-scheme loss is only 0.0580. This negative result suggests that compression sensitivity in MoEs is relational rather than purely expert-wise. ShadowMoE therefore provides both a practical compression mechanism and a framework for studying how routing topology encodes functional dependencies among experts. Anonymous artifacts: `https://anonymous.4open.science/r/topology_0723-70B3`.

1 INTRODUCTION

Mixture-of-experts models are usually viewed through the lens of sparse computation: each token activates only a small subset of experts, reducing the computation required for a large parameter count (Shazeer et al., 2017; Fedus et al., 2022). However, the routing decisions themselves expose more than sparsity. Across layers, they reveal which experts repeatedly participate in the computation of the same inputs and therefore define a structured interaction topology over experts.

This topology provides a view of model organization that is distinct from conventional parameter- or activation-level statistics. An expert is not only characterized by how often it is activated, but also by which experts precede and follow it, how strongly these transitions occur, and whether these routing patterns recur across inputs. Modern sparse systems make this structure especially observable through learned specialization, capacity-aware routing, and automatic sharding (Lepikhin et al., 2021; Du et al., 2022; Lewis et al., 2021; Zhou et al., 2022; Dai et al., 2024).

This raises a natural question: does routing topology encode information about expert functional vulnerability? In particular, if two experts receive similar amounts of traffic but participate in different cross-layer routing contexts, should they exhibit similar sensitivity to compression? We hypothesize that the answer is no. The function of an expert is partly determined by the computation paths in which it participates, suggesting that compression sensitivity may depend not only on local expert statistics but also on relational context.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

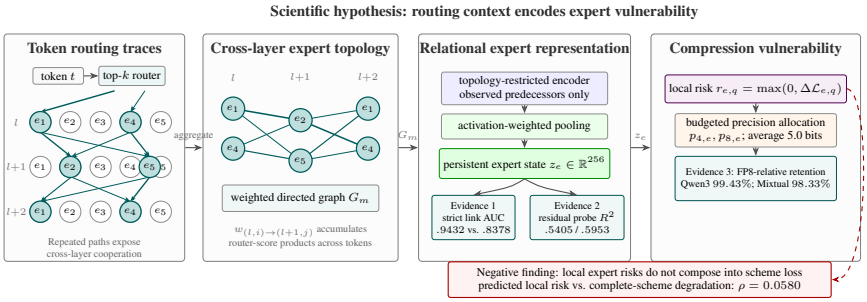


Figure 1: Scientific hypothesis and evidence chain. Token-level routes define a weighted cross-layer expert topology, from which a topology-restricted encoder learns persistent relational representations. Leakage-controlled structural prediction and quantization-residual probes test what these representations encode; budgeted INT4/FP8 allocation is a downstream application. The weak scheme-level correlation is a complementary negative finding: independently measured local risks do not compose reliably.

We introduce *ShadowMoE*, a framework for learning expert representations from routing topology. Given token-level top-k routing traces, we construct a weighted directed graph whose nodes represent layer–expert pairs and whose edges accumulate the product of router probabilities across adjacent layers. This graph preserves not only which experts interact, but also the strength of those interactions. A topology-restricted transformer then propagates information only along observed routing paths. By aggregating prompt-level expert states across a corpus of routing traces, ShadowMoE produces an expert-level representation that captures both local behavior and cross-layer context.

We evaluate the representation in stages. First, a strict held-out link-prediction task tests whether the encoder captures routing structure that cannot be recovered by directly copying the observed target edge. Second, trajectory and masked-feature probes test whether expert embeddings preserve behavioral structure. Third, quantization-residual probes assess whether the same representations contain information about compression sensitivity. Under the strict protocol, ShadowMoE improves link-prediction AUC from 0.8378 for a graph-free MLP to 0.9432, while the learned representations explain $R^2 = 0.5405$ and 0.5953 of FP8 and INT4 local quantization residuals, respectively.

We then ask whether the learned expert representations can be used to allocate precision under a global bit budget. A precision head maps each expert representation to a soft INT4/FP8 distribution, while a differentiable budget objective constrains the average bit width. At 5.0 bits on average, the resulting policy retains 99.43% and 98.33% of FP8+activation-quantized performance on Qwen3 and Mixtual V0.1, respectively.

Finally, we examine whether independently predicted expert vulnerabilities are sufficient to explain complete mixed-precision schemes. They are not. Across complete allocation schemes, the correlation between independently predicted expert risks and actual scheme-level degradation is only 0.0580. This negative result suggests that expert vulnerabilities are not additive and that jointly quantized experts can interact through the same computation topology. We therefore view ShadowMoE not simply as a quantization heuristic, but as a framework for studying relational expert vulnerability in MoE models.

Our contributions are threefold. (1) We formulate and test a routing-topology perspective on expert vulnerability through leakage-controlled structural and vulnerability probes. (2) We introduce a weighted cross-layer graph and topology-restricted encoder that transforms token-level routing traces into relational expert representations. (3) We use these representations for global-budget mixed-precision allocation that retains near-FP8 performance at 5-bit average precision, while showing that independently estimated expert risks do not reliably predict complete-scheme degradation.

2 Related Work

2.1 MoE routing as computation structure

Sparse MoE architectures reduce conditional computation by selecting a small set of experts for each token (Shazeer et al., 2017; Fedus et al., 2022). Subsequent work studies automatic sharding, capacity-aware routing, expert choice, and expert specialization (Lepikhin et al., 2021; Du et al., 2022; Lewis et al., 2021; Zhou et al., 2022; Dai et al., 2024). We build on the observation that these routing decisions also form an observable computation structure: cross-layer transitions describe which experts cooperate on the same inputs, providing context beyond activation frequency.

2.2 MoE compression and expert-wise sensitivity

Post-training and low-bit adaptation methods such as LLM.int8(), ZeroQuant, SmoothQuant, QLoRA, AWQ, GPTQ, and QuaRot address the accuracy–efficiency trade-off in large language models (Dettmers et al., 2022; Yao et al., 2022; Xiao et al., 2023; Dettmers et al., 2023; Lin et al., 2023; Frantar et al., 2022; Ashkboos et al., 2024). MoE-specific mixed-precision methods, including MxMoE, use expert-level sensitivity or local statistics to allocate precision (Duanmu et al., 2025). Our focus is complementary: we ask whether the relational context induced by routing topology contains information about vulnerability that is not captured by independent expert-wise statistics. The weak scheme-level calibration in our experiments also tests the limits of treating local sensitivity as additive.

2.3 Graph representation learning for neural computation

Graph attention and inductive graph encoders provide mechanisms for propagating information over structured neighborhoods (Velickovic et al., 2018; Hamilton et al., 2017; Xu et al., 2019; Kipf & Welling, 2017). Graph-transformer designs extend this principle to richer relational contexts (Ying et al., 2021; Rampasek et al., 2022). We use these ideas as tools for recovering relational expert context from internal computation traces; the contribution is the routing-topology formulation and its staged scientific evaluation rather than a new graph architecture. Our probing perspective is also related to work that tests what information is encoded in learned representations (Hewitt & Manning, 2019; Geva et al., 2021).

3 ShadowMoE

We consider an MoE model with L MoE layers and E experts per layer. An expert is indexed by $e = (l, i)$, and $N = LE$. For Qwen3-30B-A3B, $L = 48$, $E = 128$, and $N = 6144$. The central object is a persistent expert representation $z_e \in \mathbb{R}^{256}$ learned from the routing contexts in which e participates. A downstream precision head maps this representation to a distribution over INT4 and FP8:

$$p_e = (p_{4,e}, p_{8,e}) = \mathrm{softmax}(\alpha_e), \qquad \alpha_e = g_\phi(z_e). \tag{1}$$

The expected expert bit width and global average are

$$b_e = 4p_{4,e} + 8p_{8,e}, \qquad B = \frac{1}{N}\sum_{e=1}^{N} b_e. \tag{2}$$

For a target budget B_{target}, the differentiable policy objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{LM}} + \lambda_b (B - B_{\mathrm{target}})^2. \tag{3}$$

3.1 Routing Topology as an Expert Context

We record, for every token and layer, the top-k expert identifiers and router scores, optionally together with hidden-state, expert-output, and norm signals. For prompt m, the routing trace induces a directed graph $G_m = (\mathcal{V}_m, \mathcal{E}_m, W_m)$ whose nodes are layer–expert identities $v = (l, i)$. For adjacent layers, we accumulate the soft transition weight

$$w^{(m)}_{(l,i)\to(l+1,j)} = \sum_{t\in m} \mathbf{1}[i \in \mathcal{I}_{t,l}, j \in \mathcal{I}_{t,l+1}]\, s_{t,l,i} s_{t,l+1,j}. \tag{4}$$

Hard top-1 and unweighted top-k graphs provide reference variants. Node features combine layer/expert identity, activation count, mean routing score, routing and contribution entropy, contribution ratio, and weighted in/out degree. Edge features include w_{uv} and its row-normalized transition probability. Thus, the graph preserves both which experts interact and how strongly they interact. The graph is an explicit context for an expert, rather than a replacement for the underlying MoE computation.

3.2 Relational Expert Encoder

We seek an expert representation that is explicitly conditioned on the routing context surrounding that expert. The encoder processes each prompt graph independently and shares its parameters across prompts. It follows the Transformer attention principle (Vaswani et al., 2017), but restricts attention to graph neighborhoods as in graph attention networks (Velickovic et al., 2018). Its message-passing design is related to inductive graph encoders and graph isomorphism networks (Hamilton et al., 2017; Xu et al., 2019), while the neighborhood-restricted Transformer is consistent with graph-transformer designs (Ying et al., 2021; Rampasek et al., 2022). An expert state is updated using only its observed routing predecessors, thereby encoding computation context rather than treating experts as independent entities.

From node features to a shadow state. For node $v = (l, i)$ in prompt graph G_m, concatenate a learned layer embedding, a learned expert-index embedding, observed routing statistics, and structural statistics:

$$x_{m,v} = \left[e_l^{\text{layer}} \,\|\, e_i^{\text{expert}} \,\|\, o_{m,v} \,\|\, g_{m,v} \right], \qquad h_{m,v}^{(0)} = \text{LN}(W_{\text{in}} x_{m,v} + b_{\text{in}}). \tag{5}$$

Here $o_{m,v}$ contains activation count, mean router score, routing entropy, contribution ratio, contribution entropy, and available norm measurements; $g_{m,v}$ contains weighted in-degree, weighted out-degree, and PageRank-like statistics. Identity embeddings preserve layer–expert identity, while the remaining terms describe prompt-specific behavior. The graph-convolutional view of these features follows standard message-passing formulations (Kipf & Welling, 2017).

For block r, let $\mathcal{N}_m^-(v)$ be the predecessors of v and add a self-loop. The core update can be written compactly as

$$a_{uv}^{r,a} = \frac{\exp\Big((W_Q^{r,a} h_v^{(r)})^\top (W_K^{r,a} h_u^{(r)}) / \sqrt{d_a} \Big)}{\displaystyle\sum_{u' \in \mathcal{N}_m^-(v) \cup \{v\}} \exp\Big((W_Q^{r,a} h_v^{(r)})^\top (W_K^{r,a} h_{u'}^{(r)}) / \sqrt{d_a} \Big)}, \tag{6}$$

$$m_v^{(r)} = W_O^r \,\text{Concat}_{a=1}^{H} \left(\sum_{u \in \mathcal{N}_m^-(v) \cup \{v\}} a_{uv}^{r,a} W_V^{r,a} h_u^{(r)} \right), \tag{7}$$

$$h_v^{(r+1)} = \text{LN}\Big(\widetilde{h}_v^{(r+1)} + \text{FFN}^r(\widetilde{h}_v^{(r+1)}) \Big), \quad \widetilde{h}_v^{(r+1)} = \text{LN}\Big(h_v^{(r)} + m_v^{(r)} \Big). \tag{8}$$

Stacking R blocks summarizes increasingly long cross-layer paths and gives a final state $h_{m,e} \in \mathbb{R}^{256}$. The full query/key/value expansion and residual derivation are given in Appendix A.

3.3 Leakage-Controlled Structural Learning

We train the encoder with held-out link prediction and grouped masked-feature reconstruction. A subset $\mathcal{E}_{\text{hold}}$ of observed adjacent-layer transitions is removed before message passing; negatives are unobserved pairs from the same adjacent layer pair. The link score and combined objective are

$$\widehat{y}_{uv} = \sigma\big(h_{m,u}^\top W_{\text{link}} h_{m,v} \big), \qquad \mathcal{L}_{\text{enc}} = \lambda_{\text{link}} \mathcal{L}_{\text{link}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}. \tag{9}$$

The strict configuration is $G_m - \mathcal{E}_{\text{hold}} \to h_m \to \widehat{y}_{uv}$: held-out edges and their explicit weights are withheld from both attention and decoding. The reported AUC therefore cannot be obtained by copying the target edge feature. The probing setup follows the broader practice of testing what information is encoded in learned representations (Hewitt & Manning, 2019; Geva et al., 2021). The exact binary-cross-entropy and grouped masking operators are derived in Appendix A.

3.4 EXPERT-LEVEL AGGREGATION

After training, the encoder is frozen and run on every prompt graph. Let $h_{m,e} = f_\theta(G_m, e)$ and let $c_{m,e}$ be the number of times expert e is activated in prompt m. We use activation-weighted pooling so that prompts providing more observations of an expert contribute proportionally:

$$z_e = \frac{\sum_m c_{m,e} h_{m,e}}{\sum_m c_{m,e}}. \tag{10}$$

The purpose of this aggregation is not to estimate an expert's average activation state, but to obtain a persistent representation of the routing contexts in which the expert participates. The resulting tensor is indexed by layer and local expert identifier and has shape $[48, 128, 256]$. Unobserved experts receive a zero vector plus a missingness flag. The weighted-pooling identity and its missingness convention are formalized in Appendix A.

3.5 COMPRESSION VULNERABILITY

For expert e and quantizer $q \in \{4, 8\}$, a local pseudo-quantization experiment measures the residual

$$\Delta\mathcal{L}_{e,q} = \mathcal{L}(W_e \leftarrow Q_q(W_e)) - \mathcal{L}(W), \qquad r_{e,q} = \max(0, \Delta\mathcal{L}_{e,q}). \tag{11}$$

Only 'gate_proj', 'up_proj', and 'down_proj' are perturbed. A vulnerability predictor combines z_e, pseudo-quantization features, and activation statistics to initialize the logits in Eq. (1); its score is a ranking prior, not ground-truth end-task utility. We distinguish this local quantity from the vulnerability of a complete scheme. For a set S of jointly quantized experts and scheme quantizer Q_S, define

$$R(S) = \mathcal{L}(W \leftarrow Q_S(W)) - \mathcal{L}(W), \qquad R(S) \not\approx \sum_{e \in S} r_e, \tag{12}$$

where r_e denotes the expert-level risk score used by the allocator. The distinction is central to our negative result: local vulnerability is a useful probe target, but it is not assumed to be additive. Our quantization objective is situated relative to representative post-training and low-bit adaptation methods, including LLM.int8(), ZeroQuant, SmoothQuant, QLoRA, and AWQ (Dettmers et al., 2022; Yao et al., 2022; Xiao et al., 2023; Dettmers et al., 2023; Lin et al., 2023); the end-task table evaluates the baselines listed in Section 4.

3.6 TOPOLOGY-GUIDED MIXED PRECISION

The compression policy is a downstream use of the relational expert representation. The differentiable path constructs symmetric INT4 and dynamically scaled FP8 weights and uses their probability-weighted surrogate

$$\widehat{W}_e = p_{4,e} Q_4(W_e) + p_{8,e} Q_8(W_e). \tag{13}$$

Together with Eqs. (1)–(3), Eq. (13) yields a topology-guided static mixed-precision allocation with a global bit constraint. During strategy search the base model is frozen and only g_ϕ is trained; optional QAT jointly updates expert projections. A normalized risk prior ρ_e initializes logits as $[-2(2\rho_e - 1), 2(2\rho_e - 1)]$. The softmax and budget-penalty derivation is provided in Appendix A.

Serving integration. We additionally provide an SGLang adapter that executes the exported expert-wise INT4/FP8 policy. It dispatches each selected expert's 'gate_proj', 'up_proj', and 'down_proj' through its assigned path while preserving standard top-k routing and router-weighted aggregation. This integration demonstrates deployability, but is not evaluated as a systems-speedup contribution.

4 EXPERIMENTAL SETUP

4.1 MODELS, TRACES, AND EVALUATION

We use Qwen3-30B-A3B-base (48 routers, 128 experts per layer) and Mixtual V0.1, a Mixtral-style model activating two experts per token (Jiang et al., 2024). The Qwen3 trace corpus contains 200 prompts and 5,521 token-level traces; its soft graphs contain 487,513 node instances and 7,599,740 edges with no structural anomalies. We evaluate Winogrande, HellaSwag, PIQA, BoolQ, ARC-e, and ARC-c. Representation studies use the graph-free and hand-crafted baselines described below.

4.2 Baselines and protocol

We compare BF16, FP8+Act, INT4+Act, QuaRot (Ashkboos et al., 2024), GPTQ W4A16 (Frantar et al., 2022), MxMoE (Duanmu et al., 2025), and topology-guided ShadowMoE. Average-5.0-bit methods use the same INT4/FP8 choices. The representation evaluation separates strict held-out link prediction, trajectory similarity, and quantization-residual probes. The end-task evaluation reports accuracy for both models and compares the complete per-task results in Table 1.

4.3 Compute and scope

Model loading and smoke tests use multi-GPU A100/A800 device mapping; the code path supports eight GPUs with bounded per-device memory. The current study reports single-run measurements and does not claim multi-seed uncertainty, full-model learned QAT convergence, or fused heterogeneous-expert kernel speedups.

5 Results

We organize the evidence around four research questions: whether routing topology contains non-local structure, whether the representation contains compression information, whether it improves a fixed-budget allocation, and whether expert-wise risks compose into scheme-level loss.

5.1 RQ1: Routing topology contains non-local structure

Routing topology predicts missing transitions even when target edges and their explicit weights are withheld from the encoder and decoder. Strict soft-graph link prediction reaches AUC 0.9432, versus 0.8378 for a graph-free MLP. We exclude a leaked reconstruction result whose router-only baseline was equally strong; the strict protocol is the primary structural result. Appendix B reports the full leakage diagnosis.

5.2 RQ2: Expert representations encode compression vulnerability

The same pooled representation preserves behavioral and compression-related information. Trajectory-similarity discrimination reaches AUC 0.7026, while quantization-residual probes obtain $R^2 = 0.5405$ for FP8 and $R^2 = 0.5953$ for INT4. These probes establish that routing-conditioned states contain information about local compression residuals; they do not by themselves establish that local risk is a complete end-task surrogate.

5.3 RQ3: Topology-guided allocation improves the accuracy–bit trade-off

We use the learned states as a downstream allocation signal under a global 5.0-bit budget. Table 1 gives the complete results. ShadowMoE has the highest mean FP8-relative retention among the average-5-bit methods: 99.43% on Qwen3 and 98.33% on Mixtual, compared with $96.58\%/92.25\%$ for MxMoE. Its raw ten-metric means are 0.7248 and 0.7502, respectively. The gain is largest on routing-sensitive tasks and should be read as evidence for topology-guided static allocation, not as end-to-end learned quantization.

Functional tests confirm that SGLang loads the exported policy and completes routed MoE inference with per-expert INT4/FP8 assignments; absent fused-kernel benchmarks, this establishes compatibility rather than a systems speedup.

Table 1: End-task accuracy on Qwen3-30B-A3B-base and Mixtual V0.1. Each cell reports Qwen3 / Mixtual. Bold marks the best average-5.0-bit method on both models.

Task	Metric	BF16	FP8+Act	INT4+Act	QuaRot avg5.0	GPTQ W4A16 avg5.0	MxMoE avg5.0	ShadowMoE avg5.0
Winogrande	acc	.7230/.7735	.7269/.7758	.5375/.4775	.6815/.6709	.6798/.6547	.6859/.6714	**.7324/.7593**
HellaSwag	acc	.6231/.6516	.6204/.6528	.3047/.2755	.5831/.5794	.5789/.5668	.5914/.5871	**.6206/.6495**
	norm	.8134/.8421	.8132/.8424	.3975/.3056	.7756/.7738	.7724/.7605	.7831/.7583	**.8094/.8412**
PIQA	acc	.8101/.8210	.8090/.8270	.5990/.5457	.7762/.7758	.7769/.7791	.7835/.7811	**.8058/.8248**
	norm	.8101/.8308	.8079/.8330	.6115/.5163	.7888/.7864	.7895/.7898	.7971/.8034	**.8096/.8297**
BoolQ	acc	.8177/.8578	.8049/.8587	.3841/.3783	.7895/.7802	.7869/.7907	.8018/.7953	**.8034/.8459**
ARC-e	acc	.7980/.8388	.7988/.8380	.3413/.2694	.7724/.7708	.7687/.7627	.7786/.7614	**.7885/.8253**
	norm	.7950/.8279	.7921/.8241	.3279/.2618	.7752/.7710	.7736/.7694	.7824/.7539	**.7929/.7955**
ARC-c	acc	.5410/.5623	.5427/.5708	.2500/.2227	.5068/.5028	.5039/.4936	.5102/.5011	**.5273/.5486**
	norm	.5683/.5973	.5674/.5990	.2995/.2671	.5301/.5309	.5284/.5234	.5358/.5498	**.5580/.5819**

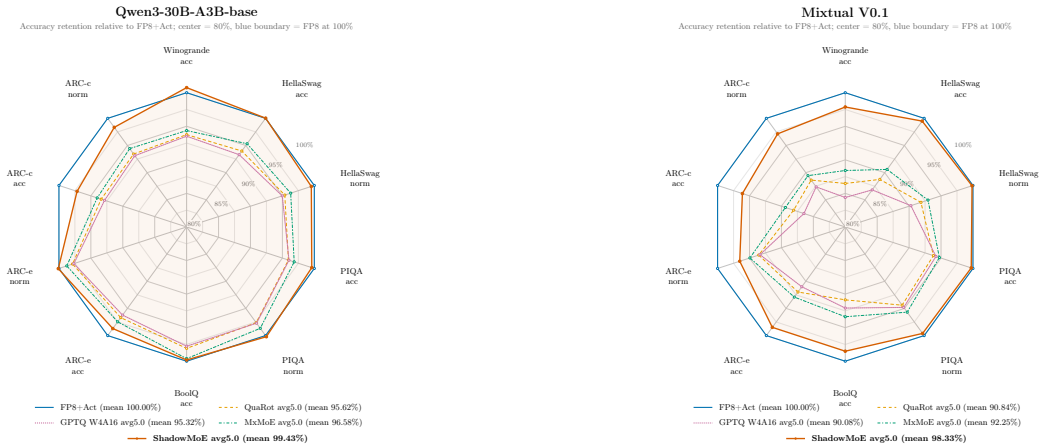


Figure 2: Accuracy retention relative to FP8+Act on Qwen3 (left) and Mixtual V0.1 (right). FP8 defines the 100% boundary; the radial range starts at 80%. ShadowMoE most closely follows the FP8 envelope on both models.

Figure 2 visualizes the retention patterns on both models.

5.4 RQ4: Expert vulnerability is not additive

Finally, we test whether independently estimated expert vulnerabilities explain complete mixed-precision schemes. They do not: the correlation between predicted local risk and actual scheme-level degradation is only 0.0580. This negative result suggests that jointly quantized experts interact through the same computation topology, so summing local risks is not a reliable model of scheme vulnerability. Appendix B provides the retrieval, allocation, and calibration diagnostics.

6 Discussion and Limitations

Our evidence supports a relational interpretation of MoE compression sensitivity: routing topology predicts structural behavior, pooled states contain information about local residuals, and the resulting policy preserves near-FP8 accuracy at a 5.0-bit average budget. The scheme-level calibration result also defines a clear boundary. The current vulnerability model operates at the individual-expert level, whereas complete schemes exhibit non-additive interactions; the allocation objective should therefore be interpreted as topology-guided static allocation rather than a complete model of joint compression loss. In addition, the study covers two MoE models and single-run main results. Broader model

and seed coverage is necessary to establish the generality of the observed relation between routing topology and compression vulnerability. The SGLang adapter demonstrates deployability, but fused heterogeneous-expert kernels and throughput are outside the present evaluation.

7 Conclusion

We show that MoE routing topology provides a relational representation of expert functionality. A weighted cross-layer graph and topology-restricted encoder predict structural behavior and compression sensitivity, enabling a static INT4/FP8 allocation that retains near-FP8 performance at a 5.0-bit average budget. The weak scheme-level calibration correlation, 0.0580, is itself a substantive finding: local expert risk does not reliably compose into complete-scheme degradation. ShadowMoE therefore offers both a practical compression application and a framework for studying relational expert vulnerability in MoE models.

AI use statement

Generative AI tools were used to assist with converting the manuscript from an earlier LaTeX format to the ICLR 2027 format and with editing and reorganizing manuscript text. The authors reviewed the resulting source, claims, citations, and rendered document, and take responsibility for the final content.

Ethics statement

Quantization can lower inference cost and emissions but may reduce reliability; safety-sensitive deployments should re-evaluate the target bit budget and disclose accuracy limits. This work uses publicly available language-model checkpoints and benchmark tasks and does not involve human subjects or personal data.

Reproducibility statement

The routing-graph construction, encoder objectives, pooling rule, allocation objective, evaluation protocol, and deployment checks are specified in the main text and Appendix B. The anonymous artifact link in the abstract provides the implementation and exported-policy details used for the reported measurements.

References

Saleh Ashkboos et al. QuaRot: Outlier-free 4-bit inference in rotated LLMs. *arXiv preprint arXiv:2404.00456*, 2024.

Damai Dai 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)*, 2024.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, 2022.

Tim Dettmers et al. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Nan Du et al. GLaM: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, 2022.

Haojie Duanmu et al. MxMoE: Mixed-precision quantization for MoE with accuracy and performance co-design. *arXiv preprint arXiv:2505.05799*, 2025.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Albert Q. Jiang et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Dmitry Lepikhin et al. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Mike Lewis et al. BASE layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, 2021.

Ji Lin et al. AWQ: Activation-aware weight quantization for LLM compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

Ladislav Rampasek et al. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Guangxuan Xiao et al. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 2023.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Zhewei Yao et al. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, 2022.

Chengxuan Ying et al. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, 2021.

Yanqi Zhou et al. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, 2022.

A Mathematical Derivations

This appendix expands the compact derivation in Section 3. The notation follows the main text: m indexes a prompt, $v = (l, i)$ indexes a layer–expert node, and $\mathcal{N}_m^-(v)$ denotes the incoming neighbors of v.

A.1 From routing scores to a weighted graph

For token t in prompt m, let $s_{t,l,i}$ be the router score of selected expert i at layer l. The indicator in Eq. (4) ensures that an edge is added only when both endpoints are selected. Expanding the sum gives the accumulated cross-layer weight

$$w_{uv}^{(m)} = \sum_{t \in m} \sum_{i,j} \mathbf{1}[u = (l, i)]\mathbf{1}[v = (l + 1, j)]\mathbf{1}[i \in \mathcal{I}_{t,l}, j \in \mathcal{I}_{t,l+1}]s_{t,l,i}s_{t,l+1,j}. \tag{14}$$

The conditional transition feature used by optional edge-aware ablations is the row normalization

$$\pi_{u \to v}^{(m)} = \frac{w_{uv}^{(m)}}{\sum_{v' \in \mathcal{N}_m^+(u)} w_{uv'}^{(m)}}, \qquad v \in \mathcal{N}_m^+(u), \tag{15}$$

where $\mathcal{N}_m^+(u)$ is the set of observed successors. Node-level structural features are computed from the same weights, for example $d_m^{\text{out}}(u) = \sum_v w_{uv}^{(m)}$ and $d_m^{\text{in}}(v) = \sum_u w_{uv}^{(m)}$. These definitions make the graph construction deterministic given the routing trace.

A.2 Full topology-restricted attention update

For completeness, the compact update in Eq. (8) is obtained by defining, for block r and head a,

$$q_v^{r,a} = W_Q^{r,a} h_v^{(r)}, \qquad k_u^{r,a} = W_K^{r,a} h_u^{(r)}, \qquad v_u^{r,a} = W_V^{r,a} h_u^{(r)}. \tag{16}$$

The attention coefficient is normalized over the incoming neighborhood plus the self-loop:

$$A_{uv}^{r,a} = \frac{\exp\big((q_v^{r,a})^\top k_u^{r,a}/\sqrt{d_a}\big)}{\displaystyle\sum_{u' \in \mathcal{N}_m^-(v) \cup \{v\}} \exp\Big((q_v^{r,a})^\top k_{u'}^{r,a}/\sqrt{d_a}\Big)}, \quad u \in \mathcal{N}_m^-(v) \cup \{v\}. \tag{17}$$

The head outputs are concatenated and projected, followed by the residual and feed-forward sublayers:

$$m_v^{(r)} = W_O^r \Big\|_{a=1}^{H} \left(\sum_{u \in \mathcal{N}_m^-(v) \cup \{v\}} A_{uv}^{r,a} v_u^{r,a} \right), \tag{18}$$

$$\widetilde{h}_v^{(r+1)} = \mathrm{LN}\Big(h_v^{(r)} + m_v^{(r)}\Big), \tag{19}$$

$$h_v^{(r+1)} = \mathrm{LN}\Big(\widetilde{h}_v^{(r+1)} + \mathrm{FFN}^r(\widetilde{h}_v^{(r+1)})\Big). \tag{20}$$

Because the normalization set contains only observed predecessors, information can propagate along observed routing paths but cannot jump to an unrelated expert. After R blocks, a projection maps $h_{m,v}^{(R)}$ to the 256-dimensional state used in the main paper.

A.3 Self-supervised losses and leakage control

Let $\mathcal{P}_m$ contain held-out positive transitions and let $\mathcal{N}_m$ contain unobserved pairs from the same adjacent-layer pair. The link loss corresponding to Eq. (9) is

$$\mathcal{L}_{\text{link}} = -\frac{1}{|\mathcal{P}_m \cup \mathcal{N}_m|} \sum_{(u,v,y) \in \mathcal{P}_m \cup \mathcal{N}_m} \left[y \log \widehat{y}_{uv} + (1 - y) \log(1 - \widehat{y}_{uv}) \right]. \tag{21}$$

For masked reconstruction, partition node features into routing ($g = \text{route}$), contribution ($g = \text{contrib}$), and norm ($g = \text{norm}$) groups. If M_g is the set of masked node/group pairs and D_g is the corresponding decoder, then

$$\mathcal{L}_{\text{mask}} = \sum_g \frac{1}{|M_g|} \sum_{(v,x)\in M_g} \|D_g(h_{m,v}) - x\|_2^2 . \tag{22}$$

The encoder minimizes the weighted sum in Eq. (9). For the strict result, each held-out edge is removed before computing all states in Eqs. (17)–(20), and w_{uv} and $\pi_{u\to v}$ are not supplied to the decoder. Thus, a high link score must be inferred from node features and the remaining topology rather than copied from the target edge.

A.4 Pooling and differentiable precision allocation

The activation-weighted pooling in Eq. (10) is a normalized weighted mean. Let $C_e = \sum_m c_{m,e}$. For $C_e > 0$, z_e is well-defined and satisfies $\sum_m (c_{m,e}/C_e) = 1$; for $C_e = 0$, we set $z_e = 0$ and append a missingness bit. This convention prevents an unobserved expert from being confused with an observed expert whose representation happens to be near zero.

For the two precision choices, $p_e = \text{softmax}(\alpha_e)$ implies $p_{4,e} + p_{8,e} = 1$ and hence

$$4 \le b_e = 4p_{4,e} + 8p_{8,e} \le 8, \qquad B = \frac{1}{N}\sum_e b_e. \tag{23}$$

The probability-weighted surrogate in Eq. (13) is therefore a convex combination of the two quantized weights. The budget term in Eq. (3) supplies a gradient to the precision logits. In particular, for $q \in \{4, 8\}$,

$$\frac{\partial b_e}{\partial \alpha_{q,e}} = p_{q,e}\,(q - b_e)\,, \qquad \frac{\partial \mathcal{L}}{\partial \alpha_{q,e}} = \frac{2\lambda_b}{N}(B - B_{\text{target}})p_{q,e}\,(q - b_e) \quad \text{(budget term only)}. \tag{24}$$

This explains why the soft policy can be optimized while its exported deterministic policy is logged at a prescribed average bit width. A lower-temperature Gumbel–Softmax sample may be used for policy selection, but the reported budget uses the deterministic probabilities in Eq. (1).

B Additional Validation

This appendix reports the intermediate experiments that support, qualify, or contradict the main-paper claims. All values are single-run diagnostics from the current pipeline. Historical representation results are retained to document why the strict protocol was introduced; they are not used as primary evidence.

B.1 Reproducibility Details

The pipeline separates trace collection, graph construction, graph-transformer training, risk labeling, static allocation, QAT, and SGLang serving. Main settings are Qwen3-30B-A3B with 48×128 experts, 200 prompts (5,521 token traces), 256-dimensional embeddings, INT4/FP8 choices, and penalty $(B - B_{\text{target}})^2$. The anonymous repository contains the exported embeddings and precision policies; full-model runs require multi-GPU A100/A800-class memory.

B.2 Limitations and Deployment Details

Full-model learned QAT remains unvalidated beyond a single-layer Qwen3 smoke test; end-task results use static INT4/FP8 allocations, all metrics are single runs, and local vulnerability labels omit joint expert interactions. The SGLang path executes the policy, but fused heterogeneous-expert kernels remain unbenchmarked, so we make no throughput or peak-memory claims. Lower-cost inference may also scale harmful use or silent reliability failures; these risks motivate target-budget evaluation and explicit disclosure of model-use limitations.

Table 2: Stage-wise validation of ShadowMoE components. Values are from the current single-run pipeline.

Component	Metric	Result
Strict graph representation	link AUC (graph / MLP)	0.9432 / 0.8378
Quantization residual probe	R^2 (FP8 / INT4)	0.5405 / 0.5953
Shadow expert similarity	AUC	0.7026
Risk retrieval	top-10 recall (5,000 labels)	0.2900
Static allocation simulator	risk avoidance (4.5 / 6 bits)	0.1902 / 0.2839
Real calibration	Spearman, predicted vs. actual loss	0.0580
Risk predictor	Spearman (1,000 labels)	0.2947
Synthetic QAT	exported average bits (target 4.5)	4.4949
Real QAT smoke test	single-layer exported average bits	4.6206

B.3 Trace Collection and Graph Integrity

The trace and graph checks establish that the representation experiments operate on valid routing data rather than malformed or incomplete graph objects. Qwen3-30B-A3B exposes 48 routers with 128 local experts each, giving 6,144 layer–expert identities. The collection pass records the selected experts and scores for every token and optionally records hidden-state, expert-output, and norm statistics.

Table 3: Model, trace, graph, and quantization-feature integrity checks. Graph counts refer to the Qwen3 trace corpus.

Object	Observed value	Validation outcome
MoE structure	48 routers $\times$ 128 experts	6,144 expert identities found
Trace corpus	200 prompts; 5,521 token traces	Collection completed
Graph variants	3 variants $\times$ 200 prompt graphs	All variants constructed
Soft routing graph	487,513 nodes; 7,599,740 edges	0 structural anomalies
Shadow embeddings	tensor $[48, 128, 256]$	Export completed
Quantization proxies	6,144 / 6,144 experts	0 NaN or Inf values

The graph anomaly checker verifies layer ordering, node identifiers, adjacent-layer edge direction, finite attributes, and valid expert indices. The zero-anomaly result is a data-integrity check, not a representation-quality metric.

B.4 Representation Evaluation and Leakage Controls

Early experiments produced very high transition-prediction scores, but a router-only baseline matched or exceeded richer feature sets. This indicated that the original target could be recovered from routing statistics without learning transferable graph structure. We therefore treat these experiments as a leakage diagnosis and use the strict protocol for the main result.

Table 4: Representation experiments and leakage diagnosis. "Historical" rows are not used as the primary graph-representation result.

Protocol or model	Metric	Result
Historical held-out transition task	area under ROC curve	0.9790
Historical class-center probe	accuracy	0.7381
Router-only reconstruction baseline	area under ROC curve	0.9795
Norm-aware reconstruction baseline	area under ROC curve	0.9553
Fifteen-feature reconstruction baseline	area under ROC curve	0.9625
Multi-task dynamic encoder	evaluation loss	0.0754
Multi-task dynamic encoder	contrastive accuracy	0.8888
Strict graph encoder	link area under ROC curve	**0.9432**
Strict graph-free multilayer perceptron	link area under ROC curve	0.8378

For strict link prediction, held-out transitions are removed before message passing and explicit transition weights and conditional probabilities are withheld from both attention and decoding. Negative pairs come from the same adjacent-layer pair. Under these controls, the graph encoder exceeds the graph-free multilayer perceptron by 0.1054 area-under-curve points. The lower strict score relative to the historical result is expected: the strict task removes the shortcuts that inflated the earlier evaluation.

B.5 Shadow-Expert Coverage and Quantization Probes

Activation-weighted pooling produces nonzero representations for 5,048 of the 6,144 layer–expert identities observed in the trace corpus. Missing experts retain a zero vector plus an explicit missingness flag. Two downstream probes assess whether the pooled embedding contains expert-level information beyond graph reconstruction.

Table 5: Shadow-expert coverage, trajectory similarity, and quantization-residual probes.

Experiment	Metric	Result
Observed shadow experts	active / total	5,048 / 6,144
High-versus-low trajectory similarity	area under ROC curve	0.7026
FP8 quantization-residual probe	coefficient of determination	0.5405
INT4 quantization-residual probe	coefficient of determination	0.5953

The similarity result shows that the pooled representation distinguishes experts with different routing trajectories. The residual probes predict quantization-related variation not fully explained by the accompanying hand-crafted statistics; they do not by themselves establish end-task utility for a joint allocation.

B.6 Risk Retrieval, Allocation, and Calibration

Risk experiments separate three questions: whether high-risk experts can be retrieved, whether a static policy avoids predicted risk under a bit budget, and whether independently predicted expert risks calibrate the loss of a complete mixed-precision scheme. Table 6 shows that these questions have materially different answers.

Table 6: Quantization-risk prediction, static allocation, and scheme-level calibration.

Experiment	Data scale	Metric	Result
High-risk expert retrieval	5,000 perturbation labels	top-10 recall	0.2900
Static ShadowMoE allocation	4.5-bit average	predicted risk avoidance	0.1902
Static ShadowMoE allocation	6.0-bit average	predicted risk avoidance	0.2839
Risk predictor	1,000 labels	Spearman correlation	0.2947
Real scheme calibration	6 allocation schemes	Spearman correlation	0.0580

The 5,000-label retrieval result supports using graph-conditioned features to prioritize expert measurements, while the 1,000-label predictor exhibits only a weak ranking signal. More importantly, the near-zero correlation over six complete allocation schemes is a negative result: summing local, independently measured expert risks does not capture interactions among simultaneously quantized experts. We therefore report static risk avoidance as a simulator diagnostic and do not present it as a surrogate for actual end-task loss.

B.7 Budget Optimization and Real-Model Smoke Tests

The differentiable policy head was first checked on 6,144 simulated experts and then wrapped around a real Qwen3 MoE layer. All experiments in the paper use only INT4 and FP8 choices. The real-model test verifies execution and gradient flow, not full-model convergence.

Table 7: Precision-policy optimization and QAT smoke-test results.

Validation	Measurement	Result
Synthetic budget search	target average bit width	4.5000
Synthetic budget search	exported average bit width	4.4949
Synthetic budget search	absolute budget error	0.0051
Real Qwen3 single-layer search	exported average bit width	4.6206
Real Qwen3 single-layer search	forward and backward passes	passed
Real Qwen3 single-layer search	policy update and checkpoint	passed

The real smoke test covers expert-projection wrapping, language-model forward propagation, backpropagation, precision-head updates, and checkpoint export. Produced artifacts include ‘policy_head.pt’, ‘precision_policy.csv’, ‘precision_policy.json’, ‘qat_model_checkpoint.pt’, ‘summary.json’, and ‘metrics.csv’. Multi-GPU loading has been exercised with bounded per-device memory on eight A100/A800-class GPUs.

B.8 SGLang Mixed-Precision Integration

The serving adapter closes the gap between an exported policy and an executable MoE model. It indexes the policy by layer and local expert identifier, selects the assigned INT4 or FP8 implementation for each expert projection, preserves SGLang’s top-k dispatch, and returns expert outputs to the standard router-weighted aggregation.

Table 8: Functional checks for the implemented SGLang mixed-precision adapter. These checks establish compatibility, not kernel speedup.

Integration check	Status
Load exported layer–expert INT4/FP8 policy	passed
Resolve layer and local expert identifiers	passed
Preserve standard top-k router dispatch	passed
Execute selected experts with assigned precision	passed
Return outputs to router-weighted aggregation	passed
Complete routed mixed-precision MoE inference	passed

These checks substantiate the deployment-compatibility claim in the main paper. They do not measure throughput, peak memory, or fused heterogeneous-expert kernel efficiency; those systems measurements remain separate work.