BEYOND ORACLE GAINS: MEASURING THE LEARNABILITY OF ADAPTIVE RAG ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) exposes multiple inference-time computational choices, including retrieval depth, generator size, and numerical precision. Oracle analyses suggest that selecting these configurations separately for each query can yield substantially better quality–cost trade-offs than using a single fixed configuration. However, an oracle observes the realized outcome of every candidate configuration and therefore does not establish whether the corresponding routing decisions are predictable before execution.

We study the distinction between *oracle opportunity* and *routing learnability* in adaptive RAG. Our analysis considers retrieval depth, generator capacity, and inference precision across four multi-hop question-answering datasets. Within a targeted four-configuration space, a retrospective oracle achieves 0.423 F1 at an average estimated cost of 36.57, while individual fixed configurations obtain only 0.203–0.258 F1. A learned router provides useful low-cost operating points but reaches only 0.249 F1 even at an average cost of 57.59.

To determine whether this gap is primarily caused by insufficient supervision, we construct 2,000 additional train-only queries. Increasing routing supervision from 1,421 to 3,421 queries improves retrieval-depth prediction by 0.072 ROC-AUC and model-size prediction by 0.042, but decreases precision-routing ROC-AUC by 0.033. These effects also vary substantially across datasets.

Our results show that large oracle headroom does not necessarily translate into an equally learnable routing policy. We argue that adaptive RAG should therefore be evaluated in terms of both available oracle opportunity and the fraction of that opportunity that can be predicted and recovered before paying the cost of candidate configurations.

1 INTRODUCTION

Retrieval-augmented generation (RAG) improves language-model inference by conditioning generation on externally retrieved evidence (Lewis et al., 2020). Most RAG pipelines nevertheless execute a largely fixed computational policy: the same number of passages is retrieved, the same generator is invoked, and the same numerical precision is used for every query.

Such uniform computation is potentially inefficient. Some questions may be answerable with one retrieved passage and a small quantized model, whereas others may benefit from additional evidence or a larger generator. This motivates *adaptive RAG*, in which the computational configuration is selected separately for each query.

We represent a RAG configuration as

$$c = (K, M, P), \tag{1}$$

where K denotes retrieval depth, M denotes generator size, and P denotes inference precision.

A natural motivation for adaptive routing is provided by a retrospective oracle. If the best configuration can be selected after observing the output of every candidate configuration, per-query selection can outperform any single fixed policy. Yet this observation leaves a critical question unanswered:

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.

> How much of the oracle opportunity is actually predictable before executing the candidate configurations?

This distinction is important because the oracle has access to information that a deployable router never observes. The oracle knows the realized answer quality under each candidate. The router must instead rely on inexpensive pre-generation signals.

Our experiments reveal a substantial gap between these two regimes. Within a targeted four-configuration experiment, the per-query oracle obtains

$$\mathrm{F1} = 0.423 \tag{2}$$

at an average estimated cost of 36.57.

The inexpensive anchor configuration, $K = 1$, Qwen2.5-0.5B, INT4, obtains only 0.203 F1 at cost 25.23. Fixed INT8 obtains 0.258 F1 at the considerably higher cost of 50.45. Thus, query-dependent configuration selection contains substantial quality–cost headroom.

However, a learned query-conditioned router remains far from this oracle. Its highest-quality operating point obtains approximately 0.249 F1 at an average cost of 57.59. Relative to the inexpensive anchor, this corresponds to recovering only a small fraction of the available oracle quality improvement.

The difficulty is not explained solely by insufficient supervision. We generate 2,000 additional non-overlapping training queries and evaluate three canonical compute transitions:

$$K : 1 \to 2, \tag{3}$$
$$M : 0.5\mathrm{B} \to 1.5\mathrm{B}, \tag{4}$$
$$P : \mathrm{INT4} \to \mathrm{INT8}. \tag{5}$$

Increasing the routing training set from 1,421 to 3,421 queries improves retrieval-depth prediction by 0.072 ROC-AUC and generator-size prediction by 0.042. In contrast, precision routing decreases by 0.033 ROC-AUC. The effect also differs substantially across datasets.

Our results therefore support a distinction between

$$\boxed{\text{Oracle Opportunity} \neq \text{Routing Learnability}} \tag{6}$$

in adaptive RAG.

Contributions. Rather than treating retrospective oracle improvement as evidence that adaptive routing is itself feasible, we explicitly separate three quantities: oracle opportunity, routing learnability, and realized oracle-headroom recovery. Our contributions are as follows.

- We study adaptive RAG as joint per-query selection over retrieval depth, generator size, and numerical precision.
- We characterize a substantial gap between retrospective oracle headroom and the quality–cost gains recovered by a deployable learned router.
- We perform a controlled scaling experiment with 2,000 additional train-only queries and show that routing learnability differs across computational axes.
- We show that scaling behavior is strongly dataset dependent and that larger training sets do not uniformly close the oracle–router gap.

2 Related Work

Retrieval-Augmented Generation. RAG augments parametric language models with externally retrieved evidence. Retrieval can improve factual coverage and provide information unavailable in

the generator parameters. However, increasing retrieval depth also increases context length, computation, and exposure to irrelevant passages. This motivates query-dependent retrieval allocation.

Adaptive Retrieval. Adaptive retrieval methods vary whether, when, or how much information is retrieved for a query. Prior work has used generation uncertainty, self-reflection, and predicted query complexity to determine when additional retrieval is warranted (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024). These approaches demonstrate that retrieval itself can be viewed as an inference-time computational decision. We extend this perspective by studying retrieval together with generator capacity and precision.

Model Routing and Adaptive Computation. Adaptive-computation systems allocate different levels of computation to different inputs. Prior work includes cascaded inference, early exiting, mixture-of-experts routing, and routing among language models of different capabilities (Teerapittayanon et al., 2016; Shazeer et al., 2017; Chen et al., 2023). Our work studies generator size jointly with retrieval depth and numerical precision rather than as an isolated model-selection problem.

Quantized Inference. INT8 and INT4 inference can substantially reduce memory and computational requirements relative to FP16 execution (Frantar et al., 2023; Lin et al., 2024). Although aggregate accuracy degradation can be small, configuration-level outcomes can vary across queries. We therefore investigate numerical precision as an additional routing axis.

Oracle-Based Adaptive Inference. Retrospective oracles are commonly used to estimate the maximum potential benefit of adaptive computation. Such analyses establish that heterogeneous configurations can be useful but do not establish that the correct decisions can be inferred before execution. Our primary focus is precisely this oracle–learnability distinction.

3 Problem Formulation

3.1 RAG Configuration

For a query q, a retriever produces a ranked list of passages

$$R(q) = (d_1, d_2, \ldots, d_{K_{\max}}). \tag{7}$$

A configuration is

$$c = (K, M, P), \tag{8}$$

where K specifies the number of retrieved passages, M specifies the generator, and P specifies inference precision.

Executing c produces

$$\hat{y}_{q,c} = G_{M,P}(q, d_1, \ldots, d_K). \tag{9}$$

Given reference answers Y_q, realized answer quality is

$$Q(q, c) = \mathrm{F1}(\hat{y}_{q,c}, Y_q). \tag{10}$$

Each configuration also incurs an estimated computational cost

$$C(q, c). \tag{11}$$

This quantity is an inference-cost proxy associated with executing a configuration rather than a hardware-independent latency measurement. For learned routing, configuration costs are normalized using statistics estimated from routing-training data only. The resulting normalized cost is used

inside the router utility, whereas Table 1 reports the corresponding average estimated inference cost on the frozen test queries.

3.2 Retrospective Oracle

The quality oracle selects

$$c_Q^*(q) = \arg\max_c Q(q, c). \tag{12}$$

Under a budget B, the corresponding constrained oracle is

$$c^*(q, B) = \arg \max_{c: C(q,c) \le B} Q(q, c). \tag{13}$$

A scalarized quality–cost formulation is

$$c_\lambda^*(q) = \arg\max_c \left[Q(q, c) - \lambda C(q, c) \right]. \tag{14}$$

The oracle is retrospective and therefore cannot be deployed directly.

Because multiple configurations can attain identical realized F1, the quality-maximizing oracle may be non-unique. We therefore distinguish the oracle quality itself from configuration identifiability: ties do not change the maximum attainable F1, but they make the identity of a single "optimal" configuration ambiguous. This ambiguity is quantified explicitly in Appendix A.

3.3 Routing Learnability

Let $x(q)$ denote inexpensive information available before expensive generation. A learned router chooses

$$\hat{c}_\theta(q) = \arg\max_c f_\theta(x(q), c). \tag{15}$$

We distinguish oracle opportunity,

$$Q(q, c^*) - Q(q, c_{\text{fixed}}), \tag{16}$$

from recovered opportunity,

$$Q(q, \hat{c}_\theta) - Q(q, c_{\text{fixed}}). \tag{17}$$

Their difference represents the oracle–router gap.

3.4 Canonical Compute Edges

The targeted scaling experiment starts from

$$c_0 = (K = 1, 0.5\text{B}, \text{INT4}). \tag{18}$$

We consider three local upgrades:

$$e_K : K = 1 \to 2, \tag{19}$$
$$e_M : 0.5\text{B} \to 1.5\text{B}, \tag{20}$$
$$e_P : \text{INT4} \to \text{INT8}. \tag{21}$$

The gain of an upgrade is

$$\Delta Q_e(q) = Q(q, c_e) - Q(q, c_0). \tag{22}$$

A positive example satisfies

$$\Delta Q_e(q) > 0.02. \tag{23}$$

4 Experimental Setup

4.1 Datasets

We use 2WikiMultihopQA (Ho et al., 2020), HotpotQA-Distractor and HotpotQA-FullWiki (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022).

The original evaluation contains 2,000 questions, with 500 questions from each dataset.

For targeted scaling, we construct 2,000 additional non-overlapping queries, again balanced across the four datasets. These queries are added only to the routing training set.

4.2 Retrieval

We use `intfloat/e5-base-v2` (Wang et al., 2022) as the dense retriever. Queries and passages are encoded using their corresponding E5 prefixes. We retrieve the top 32 passages for each query.

Our original experiments explore retrieval depths

$$K \in \{1, 2, 4, 8, 16, 32\}. \tag{24}$$

However, generation uses a 1,024-token input limit. Large retrieval depths, especially $K \geq 8$, can therefore suffer substantial context truncation. Accordingly, our primary fair routing conclusions are restricted to the low-K regime, and the targeted scaling experiment studies the clean $K = 1 \to 2$ transition.

4.3 Generators

We evaluate instruction-tuned Qwen2.5 generators (Yang et al., 2024) with multiple parameter sizes and FP16, INT8, and INT4 inference.

The targeted scaling experiment requires four configurations:

$$c_0 = (K1, 0.5\text{B}, \text{INT4}), \tag{25}$$
$$c_K = (K2, 0.5\text{B}, \text{INT4}), \tag{26}$$
$$c_M = (K1, 1.5\text{B}, \text{INT4}), \tag{27}$$
$$c_P = (K1, 0.5\text{B}, \text{INT8}). \tag{28}$$

This produces 8,000 new generation results over 2,000 additional queries.

4.4 Evaluation Metrics

We evaluate short-answer generation using Exact Match and token-level F1. Routing classifiers are evaluated using ROC-AUC and PR-AUC.

PR-AUC is particularly relevant because beneficial compute upgrades are relatively sparse.

4.5 Routing Features

The targeted routing models use inexpensive information available prior to expensive generation. Each query is represented using a normalized E5 query embedding together with lightweight lexical statistics and dataset indicators, forming a 784-dimensional representation. No candidate generation output or realized answer quality is included in the router input.

For the three canonical upgrade decisions, we train binary XGBoost classifiers predicting whether the corresponding upgrade improves answer quality by more than 0.02 F1. We select tree depth from $\{2, 3, 5\}$ using validation data only. All reported scaling results are evaluated on the same frozen test set.

4.6 Scaling Protocol

The original 2,000 queries are divided into 1,421 training, 292 validation, and 287 test examples. Because the historical split artifact was unavailable during final reconstruction, we create a deterministic dataset-stratified split with a fixed random seed and freeze this split for all targeted scaling experiments.

The additional 2,000 non-overlapping queries are used exclusively as additional routing-training supervision. They are never added to the validation or test sets. Consequently, the validation set remains fixed at 292 queries and the final test set remains fixed at 287 queries throughout the scaling experiment.

We evaluate training-set sizes of 1,421, 2,421, and 3,421 queries. Experiments are repeated with five random seeds $\{11, 22, 33, 44, 55\}$. Hyperparameter selection, including classifier tree depth, is performed using the validation set only. The frozen test set is used only for final evaluation.

4.7 End-to-End Configuration Router

In addition to the binary scaling diagnostics, we evaluate a deployable end-to-end router over the four targeted configurations. For each canonical upgrade, an XGBoost regressor predicts the realized quality gain from inexpensive pre-generation query features. These predicted gains are combined with configuration cost to select a configuration under a quality–cost trade-off.

For trade-off parameter λ, the router selects

$$\hat{c}_\lambda(q) = \arg\max_c \left[\widehat{Q}(q, c) - \lambda \widetilde{C}(c) \right], \tag{29}$$

where $\widehat{Q}(q, c)$ is the quality implied by the predicted upgrade gain and $\widetilde{C}(c)$ is the normalized configuration cost used for routing. The cost normalization is estimated from training data only.

The $\lambda = 0$ operating point prioritizes predicted answer quality, whereas positive λ values increasingly penalize expensive configurations. The reported learned-router results are evaluated on the frozen 287-query test set; test examples are not used to train the regressors or select hyperparameters.

5 Results

We ask three questions: how large is the retrospective oracle opportunity, how much of it is recovered by a deployable router, and whether additional supervision closes the gap.

5.1 Large Oracle Opportunity, Limited Router Recovery

Table 1 compares fixed configurations, representative routing policies, and the retrospective four-configuration oracle. All values in this comparison are computed on the frozen 287-query test set; no test query is used for router training or hyperparameter selection.

Table 1: **Oracle opportunity versus realizable routing on the frozen test set.** Fixed policies, learned routers, and the retrospective oracle are evaluated on the same 287 queries. The oracle retrospectively attains the highest available F1 per query; multiple configurations may attain the same maximum.

Method	F1 ↑	Cost ↓	ΔF1
Anchor	0.203	**25.23**	–
Fixed K=2	0.228	42.69	+0.025
Fixed INT8	0.258	50.45	+0.055
Fixed 1.5B	0.254	75.68	+0.051
Router ($\lambda = .15$)	0.228	34.00	+0.025
Router ($\lambda = 0$)	0.249	57.59	+0.046
Oracle$_4$	**0.423**	36.57	**+0.220**

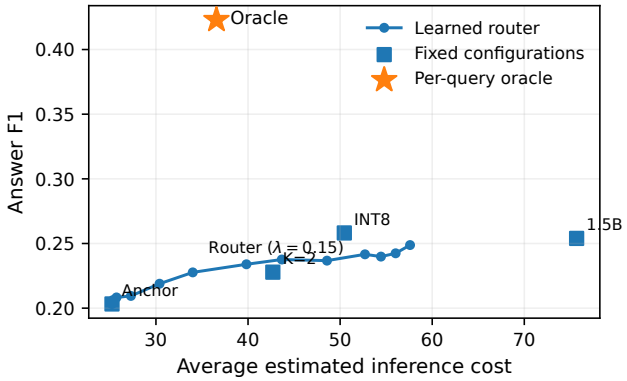

Figure 1: Quality–cost frontier. The learned router improves the low-cost region, but most retrospective oracle headroom remains unrecovered.

The oracle reaches 0.423 F1 at average estimated cost 36.57. Relative to the anchor, this corresponds to +0.220 absolute F1. The unconstrained learned router gains only +0.046, recovering approximately **21% of the available oracle headroom**.

The router nevertheless provides a practically useful low-cost policy. At $\lambda = .15$, it approximately matches fixed $K = 2$ quality while reducing average cost from 42.69 to 34.00, a **20.4% reduction**.

5.2 Additional Supervision Helps Selectively

We increase routing supervision from 1,421 to 2,421 and 3,421 training queries while keeping validation and test queries fixed.

Retrieval-depth and model-size prediction improve with additional data, whereas precision routing degrades. PR-AUC shows the same qualitative contrast for model and precision routing: 0.226→0.258 for model size and 0.235→0.185 for precision. For retrieval routing, however, the ROC-AUC improvement does not produce a corresponding material PR-AUC gain, indicating that improved global discrimination does not substantially improve ranking among the relatively sparse beneficial upgrade cases. Thus, limited sample size alone does not explain the routing gap.

5.3 Learnability Is Axis- and Distribution-Dependent

The scaling effect differs markedly across benchmarks. For retrieval routing, ΔROC is -0.025 on 2Wiki, $+0.169$ on HotpotQA-Distractor, $+0.086$ on HotpotQA-FullWiki, and $+0.101$ on MuSiQue. For model routing, the corresponding changes are -0.036, $+0.109$, -0.005, and $+0.173$. Precision routing shows no consistent positive scaling.

The broader configuration diagnostic also reveals substantial target degeneracy: 89.95% of queries admit multiple quality-equivalent optima, the median number of tied optimal configurations is 12, and the median oracle margin is zero. These observations indicate that routability is jointly determined by the compute axis, data distribution, and observability of the relevant signal.

Table 2: **Targeted supervision scaling.** ROC-AUC is averaged over five seeds; the final column gives the paired seed-bootstrap 95% interval for the 1,421-to-3,421 change. Reported differences and confidence intervals are computed from unrounded seed-level values; displayed ROC-AUC values are rounded to three decimals.

Axis	1,421	2,421	3,421	ΔROC (95% CI)
Retrieval	0.478	0.540	**0.551**	**+0.072** [+.063,+.080]
Model	0.520	0.536	**0.563**	**+0.042** [+.024,+.061]
Precision	**0.591**	0.579	0.558	**-0.033** [-.059,-.012]

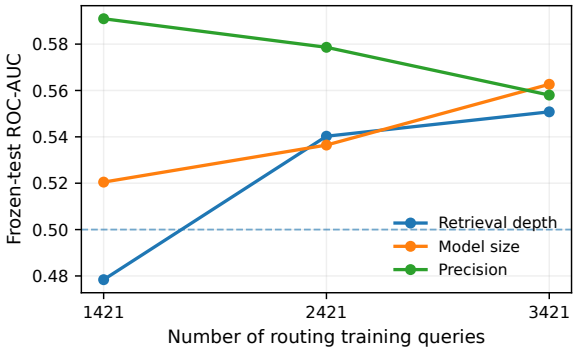

Figure 2: Frozen-test routing performance as training supervision increases.

6 Discussion

Oracle opportunity is not routing learnability. A large retrospective oracle establishes that adaptive computation is useful in hindsight, but it does not establish that the required decision is predictable before inference. Our results expose a large gap between these quantities.

Why does the gap persist? Beneficial upgrades are sparse, their average gains are small, and many oracle outcomes are tied. In addition, the utility of extra retrieval or model capacity may depend on retrieved evidence that is not fully observable from the query representation.

Not every compute axis should be routed. Retrieval depth and model capacity improve with additional supervision, whereas numerical precision does not. Under the feature set, generator family, and quantization setting studied here, this suggests that some inference dimensions may be better treated as fixed deployment choices rather than query-conditioned actions.

Precision upgrades provide a particularly weak routing signal. The additional training-query diagnostic helps explain the different scaling behavior across compute axes. The INT4-to-INT8 transition has both the lowest positive-upgrade rate (15.3%) and the smallest mean quality gain (+0.0038 F1), compared with 20.0% and +0.0121 for retrieval depth and 22.25% and +0.0123 for model size. Thus, the precision decision provides substantially less observable quality-improvement signal for a query-conditioned router.

Practical value remains. Incomplete oracle recovery does not make routing useless: our router approximately matches fixed $K = 2$ quality at 20.4% lower estimated cost. Adaptive RAG should therefore report both oracle headroom and the portion actually recovered by a deployable policy.

7 Limitations

We study Qwen2.5 generators, E5-base-v2 retrieval, multi-hop QA, and token-level F1; other models, retrievers, tasks, and metrics may induce different routing landscapes. Our cost measure is an estimated relative inference proxy rather than direct wall-clock latency, so the reported quality–cost frontier should be interpreted as a controlled comparison of relative computational demand rather than a hardware-independent latency benchmark.

The final scaling experiment evaluates four canonical configurations and uses a deterministic reconstructed 1,421/292/287 split because the historical split artifact was unavailable. The 1,024-token input limit also affects large retrieval depths, so our primary retrieval-routing conclusions are restricted to the low-K regime. Richer retrieval-aware or execution-aware signals may recover more oracle headroom.

8 Conclusion

We studied adaptive RAG through the gap between retrospective oracle opportunity and realizable routing. Although substantial per-query configuration headroom exists, learned routing recovers only part of it. More supervision improves retrieval-depth and model-size routing but not precision routing.

Our results therefore suggest a stricter criterion for adaptive inference: *oracle opportunity, routing learnability, and recovered headroom should be measured separately*.

AI Use Statement

Generative AI tools were used to assist with code development and debugging, LaTeX formatting, language editing, and improving the clarity and organization of the manuscript. They were also used as an interactive aid during exploratory analysis and interpretation of experimental results.

All experimental designs, methodological decisions, data processing procedures, reported numerical results, and scientific claims were reviewed and verified by the authors. AI-generated code and manuscript edits were inspected and, where applicable, tested against the underlying experimental outputs.

The authors take full responsibility for the final content of this work, including all text, claims, analyses, and artifacts produced with the assistance of generative AI tools.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 7036–7050. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.389.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe

Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition*, 2016.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259.

A Additional Diagnostics

Table 3: Oracle ambiguity in the broader configuration diagnostic.

Statistic	Value
Unique oracle optimum	10.05%
Multiple oracle optima	89.95%
Mean tied configurations	11.9
Median tied configurations	12
Mean oracle margin	0.0384
Median oracle margin	0.0000
Cheapest configuration exactly optimal	48.25%
Cheapest configuration within 0.05 F1	48.70%
All configurations obtain zero F1	30.85%

Table 4: Configuration-sensitivity taxonomy over the original 2,000 queries.

Category	Count	Fraction
Cheap sufficient	357	17.85%
Retrieval-sensitive	44	2.20%
Model-sensitive	70	3.50%
Precision-sensitive	14	0.70%
Multi-axis	898	44.90%
Unsolvable	617	30.85%

Table 5: Canonical local-upgrade statistics on the additional 2,000 routing-training queries. Positive means ΔF1 > 0.02 relative to the anchor. These queries are used only as additional router supervision and never enter the frozen validation or test sets.

Upgrade	Positive	Rate	Mean ΔF1
$K : 1 \rightarrow 2$	400	20.00%	+0.0121
0.5B$\rightarrow$1.5B	445	22.25%	+0.0123
INT4$\rightarrow$INT8	306	15.30%	+0.0038