Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deterministic multi-hash routing supports long-horizon training in a compact language model

Boris Peyriguere

peyriguere.boris@gmail.com

Complexity-ML

Article

Keywords:

Posted Date: August 28th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10788774/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Deterministic multi-hash routing supports long-horizon training in a compact language model

Boris Peyriguere
Independent researcher, Complexity-ML
Paris, France
Corresponding author: Boris Peyriguere
peyriguere.boris@gmail.com

Abstract

Mixture-of-experts language models usually learn a router from contextual representations. We investigate whether expert selection can instead be fixed by token identity while attention remains contextual. We train a 201.2-million-parameter decoder whose feed-forward blocks combine an always-active shared branch with two of four narrow residual experts selected from persisted, layer-specific multi-hash tables. The model undergoes 129.996 billion base-token exposures, 32.069 billion full-parameter refinement exposures and three epochs of full-parameter instruction tuning. With zero-shot causal-choice scoring on complete public splits, the final checkpoint obtains 68.01% PIQA accuracy, 57.24% ARC-Easy, 27.13% ARC-Challenge and 33.21% HellaSwag. Its 47.29% Combined ARC score exceeds locally evaluated public decoders containing 350–774 million parameters under the same scoring protocol. These results establish long-horizon trainability and practical auditability, but do not isolate routing as the cause because no matched full-scale dense control or multi-seed replication is available. We release the checkpoint, routing tables, training code, evaluation outputs and serving implementation to support independent replication.

Sparse mixture-of-experts (MoE) models increase stored capacity while activating only part of each feed-forward block. Most successful systems learn a router from the current hidden representation and add capacity constraints or balancing objectives to keep expert traffic tractable [1–4]. This design is expressive, but expert selection changes during optimization and depends on both the current activation and router parameters. Routing behaviour therefore has to be measured after execution rather than known in advance.

Fixed lexical routing offers a different trade-off. Hash Layers showed that token identity can provide a stable routing signal without a learned gate [5]. A fixed route is reproducible before training, its load can be audited directly from the table and inference does not evaluate a router network. The central concern is also immediate: the same token receives the same expert pair regardless of context, so fixed routing may constrain useful representation learning.

Here we examine this question at the scale of a complete compact language-model lineage rather than a short architecture probe. We report the architecture, training exposure,

Table 1: Architecture and training lineage. Token figures denote exposures, not distinct documents. The refinement stage stopped at step 8,156 of 17,802.

Item	Value
Trainable parameters	201,194,368
Layers / hidden size	16 / 896
Query / key-value heads	14 / 2
Vocabulary / context	32,000 / 2,048
Shared width	3,072
Stored experts / expert width	4 / 64
Active experts	top-2
Independent hash channels	2
Base-token exposure	129.996B
Refinement exposure	32.069B
Instruction-tuning set	300,000 examples
Instruction-tuning epochs	3

route audit and full-split zero-shot evaluation of TR-HASH MoE 200M, a 201.2-million-parameter decoder trained through base pretraining, an interrupted but evaluated full-parameter refinement stage and full-parameter instruction tuning. Our claim is deliberately bounded: fixed multi-hash routing remains trainable and produces a competitive compact checkpoint. We do not claim that routing caused the observed quality advantage.

Results

A shared contextual path with fixed residual routes

The model contains 16 pre-normalized decoder layers, hidden size 896, rotary position embeddings [6], grouped-query attention with 14 query heads and 2 key-value heads [7], tied input-output embeddings and a 32,000-token vocabulary. Each feed-forward block adds a shared SwiGLU path of width 3,072 to two routed residual experts of width 64. Four experts are stored per layer. The active intermediate width is therefore 3,200 while the stored width is 3,328 (Table 1).

For token identity t at layer l, a persisted integer table $R_l(t)$ returns two distinct experts. If S_l is the shared transformation and $E_{l,e}$ is expert e, the block computes

$$F_l(x,t) = S_l(x) + \sum_{e \in R_l(t)} E_{l,e}(x). \tag{1}$$

Attention has already made x contextual; only the residual capacity choice is lexically fixed. Expert projections are stored as stacked tensors so that expert identity and route-table identity remain explicit in checkpoints.

Table 2: Matched held-out instruction-tuning evaluation. Each epoch contains 1,994 optimizer steps.

Checkpoint	Step	Loss	Perplexity
Source	0	1.722610	5.60
Epoch 1	1,994	0.990943	2.69
Epoch 2	3,988	0.963912	2.62
Epoch 3	5,982	**0.959617**	**2.61**

Multi-hash compilation produces auditable routes

For every layer, each token-expert pair receives two independent deterministic 32-bit rendezvous scores. Summing the channel scores and selecting the two highest-scoring distinct experts compiles a fixed table of shape $[2, |V|]$. Across all 16 layers, no token selects the same expert twice. First-layer expert occupancy differs by less than 0.7% from the uniform vocabulary-level target. This is a statement about table construction, not runtime token-frequency balance: corpus frequencies can still produce unequal traffic.

Persisted routes remove an otherwise hidden moving part from the checkpoint. The route for a token can be inspected without forwarding an activation, and a checkpoint hash covers both expert weights and the integer routing tables. A first-step parameter-update canary additionally checks embeddings, attention, shared feed-forward weights and routed expert tensors. This guard detected silent loss of autograd when an optimized inference kernel was mistakenly used during training.

Quality improves through refinement and full-parameter tuning

Base training exposes the model to 129,995,636,736 tokens and ends with a minibatch loss of 2.6526 and perplexity 14.19. A fresh optimizer and learning-rate schedule then revisit the same 70-billion-token unique selection without replay. The refinement is interrupted after 32.069 billion additional exposures rather than represented as a completed 70-billion-token pass.

Instruction tuning starts from the refinement checkpoint and updates every parameter for three epochs. The training set contains 300,000 examples and the matched held-out set 3,000 examples. Sequence packing converts the training examples into 127,554 packed sequences per epoch with 77.7% payload utilization. Held-out loss falls from 1.7226 before tuning to 0.9596 after the third epoch (Table 2).

PIQA supplies a consistent checkpoint-level probe across the lineage. Raw accuracy rises from 65.45% at the base endpoint to 68.66% at refinement step 8,156, while the final instruction checkpoint retains 68.01%. Normalized accuracy is highest after instruction tuning at 69.10%. The refinement gain is therefore not erased by full-parameter post-training.

Table 3: Zero-shot final-checkpoint evaluation on complete public splits.

Benchmark	Examples	Accuracy	Normalized
PIQA	1,838	68.01%	69.10%
ARC-Easy	2,376	57.24%	52.86%
ARC-Challenge	1,172	27.13%	29.01%
ARC Combined	3,548	**47.29%**	44.98%
HellaSwag	10,042	33.21%	38.74%

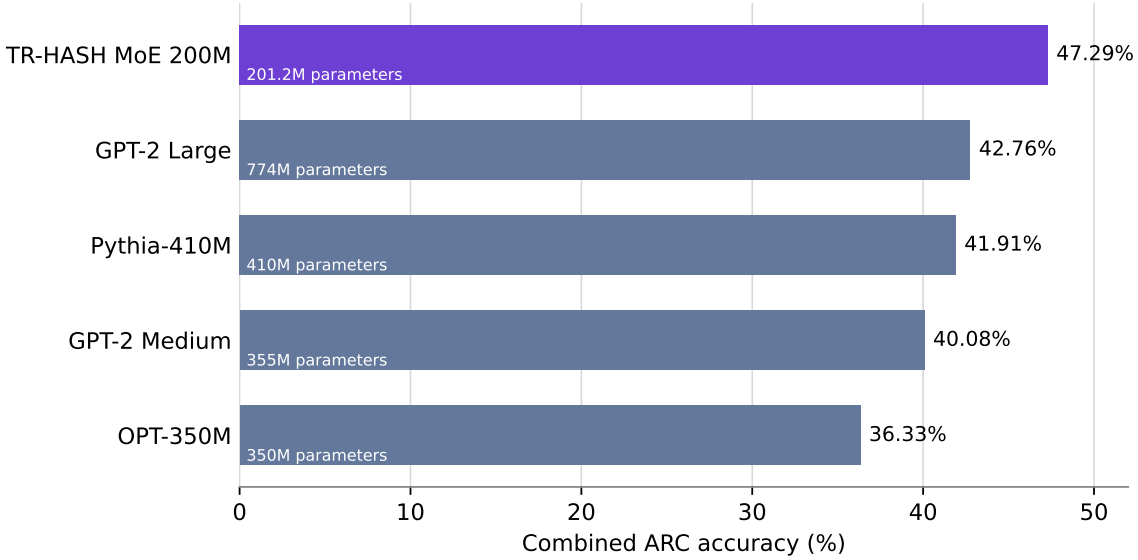


Figure 1: Combined ARC zero-shot micro-accuracy under one documented causal-choice protocol. All 2,376 ARC-Easy and 1,172 ARC-Challenge examples are included. Baselines are public dense decoders evaluated locally with the same prompting and aggregation. The comparison is not matched for data, tokenization, optimization or training budget.

Full-split zero-shot evaluation

We evaluate every example in the public PIQA validation, ARC-Easy test, ARC-Challenge test and HellaSwag validation splits as a causal continuation, without demonstrations, a system message or a chat template. The candidate with the largest summed conditional log-likelihood determines accuracy; normalized accuracy divides by candidate token count.

The final checkpoint obtains 68.01% PIQA accuracy, 57.24% ARC-Easy, 27.13% ARC-Challenge and 33.21% HellaSwag (Table 3). Combined ARC is the micro-average over all 3,548 ARC examples, not the arithmetic mean of the two percentages. The resulting 47.29% exceeds four locally evaluated public dense decoders containing between 350 million and 774 million parameters under the same prompt construction and scoring rule (Fig. 1). This comparison provides parameter-efficiency context; it is not a controlled architecture experiment because training data, tokenizers, optimization and token budgets differ.

Discussion

The experiment demonstrates that context-independent expert selection need not prevent a compact decoder from sustaining a long training lineage. Fixed multi-hash routes remain numerically stable through base training, a fresh optimizer refinement and full-parameter instruction tuning. The final model also produces non-trivial results on physical reasoning, science questions and commonsense continuation tasks. Together with persisted route tables and duplicate-free top-2 assignments, this makes deterministic routing a viable object of controlled study rather than only an implementation curiosity.

The evidence does not establish architectural superiority. There is one training lineage, no full-scale parameter-matched dense or learned-router control, no multi-seed replication and no completed benchmark-contamination audit. External models differ in their corpora and token budgets, and the target and baselines are evaluated through two numerical backends. Instruction tuning can also alter multiple-choice likelihoods without producing reliable open-ended assistance. Claims are consequently limited to trainability, auditability and measured checkpoint performance.

The most informative next experiment is a pre-registered matched comparison of a dense residual block, a learned top-2 router and deterministic multi-hash routing. Parameters, active width, data order, optimizer, token budget and evaluation code should be held constant, with multiple seeds at a scale where replication is affordable. Such a study would test whether fixed routing changes optimization or generalization rather than merely showing that it can work.

Methods

Model and routing

The decoder uses RMS normalization, rotary positions, grouped-query causal attention and SwiGLU feed-forward transformations. For each layer and token, two independent 32-bit score channels rank the four experts. The two highest distinct summed scores are stored in an integer table. Route construction is deterministic from fixed seeds and is independent of model parameters.

The shared path is always evaluated. Routed gate, up and down projections are stored as three-dimensional tensors indexed by expert. During execution, token rows are grouped by their two table-selected experts, projected through the corresponding slices and added to the shared output. No router logits, softmax, capacity factor or auxiliary balancing loss is used.

Training lineage

Base pretraining selects a 70-billion-token unique corpus and applies source-specific replay counts to reach 129.996 billion exposures. Sources include filtered web text, educational web text, code, mathematics and synthetic textbook-style data. The base run completes 165,298 optimizer steps.

Refinement initializes from base weights only, with fresh optimizer state and a new schedule. It revisits the same unique-token selection without augmentation or replay and

stops at step 8,156 of 17,802. Instruction tuning then initializes from the refinement weights, uses a 300,000-example mixture with a 3,000-example matched holdout and trains all 201.2 million parameters for three epochs.

Multiple-choice evaluation

For a context c and candidate continuation $y = (y_1, \ldots, y_m)$, summed score is

$$s(y \mid c) = \sum_{i=1}^{m} \log p(y_i \mid c, y_{<i}). \tag{2}$$

Normalized score is $s(y \mid c)/m$. The selected candidate maximizes the relevant score. Inputs are capped at the native 2,048-token context. No chat template, instruction wrapper or few-shot demonstration is used.

The target, GPT-2 and Pythia checkpoints are evaluated in FP16 through one array backend. OPT-350M is evaluated in FP16 through PyTorch because the first backend does not implement OPT. Token-level scoring, split selection and result aggregation are otherwise identical. Backend differences remain a limitation.

Statistics and reproducibility

Reported benchmark values are deterministic point estimates over complete public splits, so no sampling confidence interval is attached. They do not measure variation across training seeds. The submission package contains the model configuration, exact machine-readable benchmark outputs and a manifest of the public evaluation, model, data and implementation artifacts. The public repositories additionally provide evaluation scripts, route-table generation code, route audits and parity fixtures.

Data availability

The benchmark datasets are publicly available from their original sources. The processed 300,000-example instruction-tuning mixture and its tokenizer are available in the public SFT dataset repository. The released checkpoint, base checkpoint and refinement checkpoint are archived separately. Upstream documents retain their original licenses and terms.

Code availability

Model construction, deterministic routing, training and evaluation code are available in the public framework repository. The serving implementation is also public. Exact benchmark outputs and the artifact manifest are included with the submission. Each public repository states its applicable license.

Author contributions

B.P. conceived the architecture, implemented the model and training and evaluation software, conducted the experiments, analysed the results and wrote the manuscript.

Funding

This research received no external funding. Computing resources were self-funded.

Competing interests

The author declares no competing interests.

Use of generative AI tools

Generative AI tools assisted with language editing, consistency checks against author-provided measured artifacts and document preparation. The human author supplied the ideas and results, verified every numerical claim and takes responsibility for the manuscript.

References

[1] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of ICLR*, 2017.

[2] 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.

[3] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Thibaut Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[4] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of ACL*, 2024.

[5] Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, 2021.

[6] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[7] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of EMNLP*, 2023.