TR-Hash 200M: Multi-Hash Token-ID Routing for Shared Residual Experts

Architecture and Training Plan

Boris Peyriguere
Complexity-ML, Independent Research
boris@complexity-ai.fr

August 2026

Status: training in progress. This report documents the realized architecture, a training-independent route-table audit, and the data replay schedule. It reports no pretraining loss, evaluation, or downstream metric — those require the run to finish and will follow in an updated report.

Abstract

We document the architecture and training plan of a second TR-Hash system: a 201.2M-parameter decoder in which token identity deterministically selects two of four small residual experts, while every token retains an always-on shared SwiGLU path. This report extends our earlier 492.1M/20B system in two respects: routing uses a multi-hash rendezvous scheme (`token_id_multi_hash`, two independent 32-bit hash channels summed per token/expert pair) rather than a single hash plus derived stride, and the shared-to-expert width ratio is substantially wider (shared width 3,072 against expert width 256, versus 4,864 against 128), so experts hold a smaller fraction of total capacity (5.47% here against a larger share in the earlier system). The route tables are training-independent — fixed at model construction from the token vocabulary and a seed, not learned — so we audit them now, before training completion: across 16 layers, all route tables are pairwise distinct, no token selects the same expert twice, and per-layer expert occupancy is marginally balanced (15,917–16,118 of 64,000 slots per expert on the first layer). The model is training on a replay-scheduled mixture drawn from a 200B-token corpus (DCLM, FineWeb-Edu-Dedup, Stack-Edu, FineMath, InfiWebMath, Cosmopedia v2): 70B unique tokens, replayed with source-specific pass counts to a 130B-token exposure budget. As of this report, training has not completed; we publish the architecture and schedule ahead of results for the same reason the earlier report published its checkpoint audit separately from its claims — so the specification is checkable independent of any outcome.

1 Introduction and motivation

Mixture-of-experts (MoE) language models commonly learn a contextual router that selects parameters for each hidden state [8, 3, 4]. This is expressive, but it also entangles several questions. A quality change may come from additional expert capacity, a better contextual decision, router optimization, load-balancing objectives, or the dispatch implementation. Routes may drift during training and can collapse without balancing controls.

Fixed token-ID routing is a deliberately restricted alternative. It asks whether stable lexical identity can address a small residual parameter subspace while attention and a dense shared MLP continue to carry context. The route is reproducible from a persisted table, requires no learned gate, and can be audited independently of model activations or training progress. Hash Layers demonstrated that fixed lexical hashes can be useful routing signals [6]; shared-expert architectures separately motivate keeping common computation active for every token [2, 5].

Our earlier 492.1M-parameter report used a single hash plus a derived stride to build each layer's route table. This report documents a second, independent test of the same design philosophy at a smaller scale (201.2M parameters) with two deliberate changes: a multi-hash rendezvous route builder, and a much wider shared-to-expert ratio. Its contributions are:

1. an exact description of the realized 201.2M shared-plus-residual architecture, including the multi-hash route construction;
2. a route-table audit performed independently of training completion, since routes are fixed at construction;
3. the exact data replay schedule (70B unique tokens, source-specific pass counts, 130B-token exposure) the model is currently training on; and
4. an explicit statement of what this report does *not* yet claim: no pretraining trajectory, held-out evaluation, or downstream task result is included.

2 Architecture

2.1 Contextual backbone

The model is a pre-norm causal decoder with rotary position embeddings, RMSNorm, query/key normalization, tied input/output embeddings, and grouped-query attention (GQA) [1, 7]. It has 16 layers, hidden size 896, 14 query heads, two key/value heads, vocabulary size 32,000, and maximum position length 2,048. Deterministic routing changes only the feed-forward sublayer; it does not replace attention or contextual hidden states.

2.2 Shared and routed computation

Let x_l be the contextual hidden state at layer l, t a token ID, S_l the shared SwiGLU, and $E_{l,e}$ one of four residual SwiGLU experts. A fixed table $R_l \in \{0,1,2,3\}^{2\times 32{,}000}$ supplies two routes $R_{l,1}(t) \neq R_{l,2}(t)$. The feed-forward output is

$$F_l(x_l, t) = S_l(x_l) + 2\sum_{j=1}^{2} 0.5\, E_{l,R_{l,j}(t)}(x_l). \qquad (1)$$

The shared branch has width 3,072. Four stored experts each have width 256, giving stored intermediate width $3072 + 4 \times 256 = 4096$. Two experts are active, so active intermediate width is $3072+2\times256 = 3584$. The exported configuration records shared scale one, routed scale two, top-k = 2, and fixed primary weight 0.5 – the same combination rule as our earlier report, applied to a wider shared branch and narrower experts (5.47% of parameters are expert tensors here, against a larger share previously).

2.3 Multi-hash route construction

Unlike a single hash plus derived stride, each layer's route table is built by rendezvous hashing: every (token ID, expert) pair receives `route_hash_count`= 2 independent 32-bit scores, one per hash channel; the channel scores are summed, and the top-k experts by summed score become that token's route. This makes the final route depend on multiple independent hash draws rather than one hash and a derived offset, while still compiling to the same [top-k, vocab] table consumed by the fused dispatch kernel – there is no additional runtime routing cost relative to the single-hash construction.

2.4 Route-table audit

Route tables are fixed at model construction from the vocabulary size, expert count, and a seed; they do not depend on training progress, so we audit them now rather than waiting for a trained checkpoint. Direct inspection of the constructed model finds:

- all 16 layer route tables are pairwise distinct;
- no token selects the same expert twice within its top-2, for every layer;
- per-layer expert occupancy is marginally balanced: the first layer's four experts occupy 15,954 / 16,118 / 15,917 / 16,011 of 64,000 route slots ($2 \times 32{,}000$), each within 0.7% of the uniform 16,000 target.

As with our earlier report, this establishes marginal balance, not uniformity over every ordered expert pair; we do not claim the ordered-pair distribution is uniform.

3 Data and replay schedule

Source	Unique tokens	Passes
DCLM	20B	1
FineWeb-Edu-Dedup	25B	2
Stack-Edu	8B	2
FineMath	5B	3
InfiWebMath	5B	3
Cosmopedia v2	7B	2
Total	**70B unique**	**130B trained**

Table 1: Replay plan selected from the immutable 200B-token source corpus: 70B unique tokens, replayed with source-specific pass counts to a 130B-token training exposure.

The 70B unique tokens are selected without copying or retokenizing any shard from the immutable 200B-token source corpus; the replay plan is a manifest of shard selections and per-source pass counts, verified before training against source manifests, referenced shard existence, shard sizes, and content digests. Training tokenizes with the project's 32K vocabulary, matching the architecture's vocabulary size.

4 Status and limitations

This report intentionally stops short of a pretraining record. As of publication, the run has not completed, and we do not report a held-out loss trajectory, a final step count, or any downstream evaluation – reporting placeholder or partial numbers for an unfinished run would misrepresent the state of evidence. What *is* reported here – exact parameter counts, the route construction and its audit, and the exact data schedule – is independent of training completion and will not change when training finishes.

An updated report will add the pretraining trajectory, final checkpoint statistics, and any downstream sanity checks once the run completes, following the same evidence conventions as our earlier report: single seed, single architecture, single data mixture, no matched dense or learned-router baseline, and no claim of superiority over either alternative.

5 Conclusion

TR-Hash 200M is a 201.2M-parameter architecture specification and an in-progress 70B-unique/130B-token pretraining run. The multi-hash route construction and the shared/expert width ratio differ deliberately from our earlier 492.1M report, and both are documented and audited here independent of training outcome. This report makes no pretraining or capability claim; it exists so the architecture and data schedule are checkable before results arrive, not only after.

Reproducibility. Implementation and tests are maintained at https://github.com/Complexity-ML/complexity-framework. The replay-plan builder, route audit, and model configuration referenced here are part of the tracked source.

Use of generative AI tools. Generative AI tools assisted with language editing, consistency checks against author-provided configuration and measured route/parameter facts, and packaging. The author reviewed the numerical results, claims, source, and final text and takes responsibility for the work.

References

[1] 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. *arXiv preprint arXiv:2305.13245*, 2023.

[2] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[3] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

[4] 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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[5] Huy Nguyen, Thong T. Doan, Quang Pham, Nghi D. Q. Bui, Nhat Ho, and Alessandro Rinaldo. On deepseekmoe: Statistical benefits of shared experts and normalized sigmoid gating. *arXiv preprint arXiv:2505.10860*, 2025.

[6] Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. *Advances in Neural Information Processing Systems*, 34:17555–17566, 2021.

[7] 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.

[8] 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. *arXiv preprint arXiv:1701.06538*, 2017.