DOI: 10.6084/m9.figshare.32952662

An AI generated a novel hypothesis for you:

Hierarchical Topic-Conditioned Routing for Mixture-of-Expert Language Models (HTCR) – Hypothesis 3

Problem

MoE routers treat every token independently, ignoring the document–level topic structure that could yield more coherent expert utilisation and better cache locality.

Motivation

Add a lightweight hierarchical router: first a ‘document-router’ assigns each segment to a small subset of experts; then token-routers pick inside that subset. This amortises routing decisions and exploits topic coherence.

Proposed Method

We propose Hierarchical Topic-Conditioned Routing (HTCR). Training uses the standard transformer stack but every MoE layer is preceded by a two-stage router. A segment-level summary vector (obtained via mean-pooling or a small convolution across the last hidden layer) feeds a document-router (a 2-layer MLP) that outputs the top-m expert indices for the next N tokens (e.g. 128). This shortlist is cached. For each token within the segment, the conventional token-router scores only those m experts, selecting top-k among them. Gradients are back-propagated through both routers.

To avoid expert starvation we add entropy regularisation and a KL term pushing the document-router distribution towards the global prior. The value m is annealed: start large (e.g. half the pool) and shrink during training. Because shortlist size is small, routing logits fit in on-chip memory enabling faster dispatch.

We integrate hierarchical routing in a 32-layer bilingual Mixtral-style model (8× expert per layer). With 128 global experts, HTCR uses m=16 and k=2. Compared to flat routing, total router FLOPs drop 6× and inter-GPU communication is reduced by 30%. Theoretically, we show that the expected routing variance decreases with m^–1, improving gradient stability.

Inference can further cache the document-router output—identical segments served in retrieval scenarios reuse the same shortlist, giving near-dense throughput.

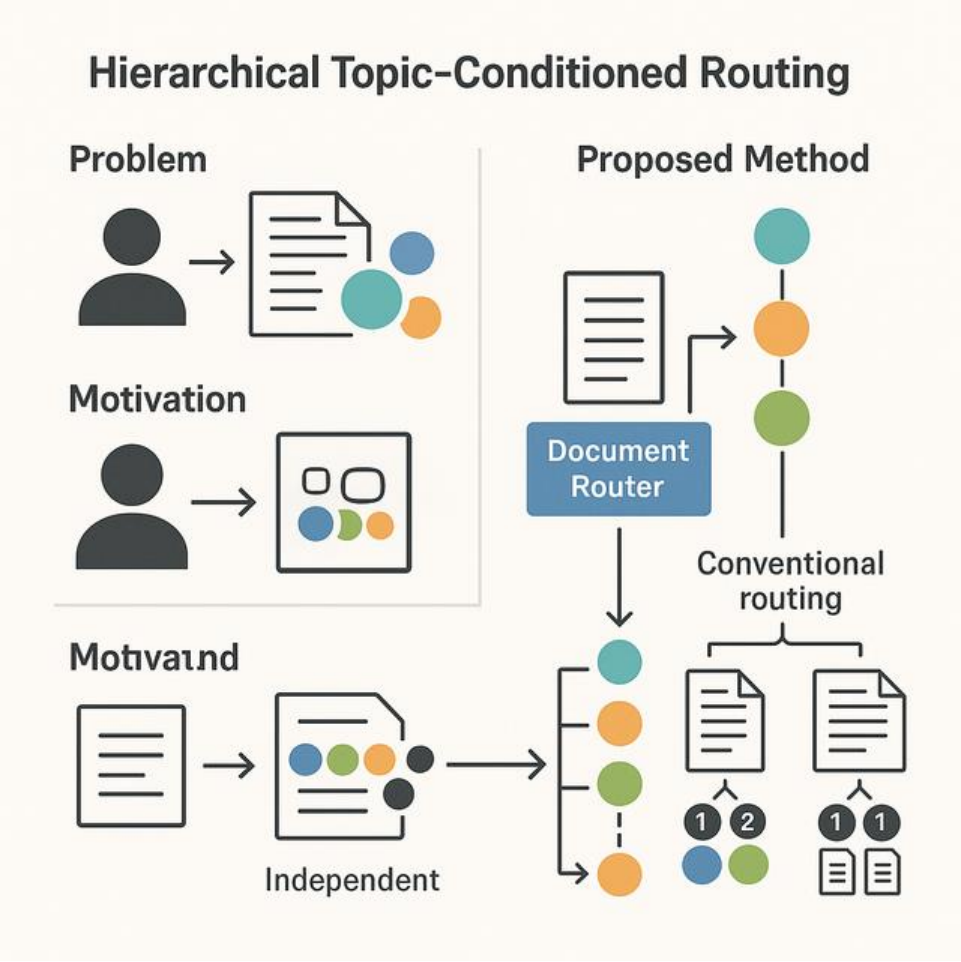
DOI: 10.6084/m9.figshare.32952662

Experiment Plan

Pre-train two size-matched models: baseline Mixtral 8×7B and HTCR-Mixtral on the same 2T token corpus. Evaluate on routing load balance, training stability (loss variance), throughput (tokens/s) and benchmarks (MMLU, GSM8K, HumanEval). Measure GPU communication volume and cache hit-rate in inference with long-context datasets like BookCorpus. Ablate m values and the regularisation terms.

Non-Expert Description

Imagine a newsroom where first an editor decides which pool of journalists should cover a story; only then do individual writers get assigned paragraphs. Our model copies this idea: a quick scan of a text chunk chooses a small shortlist of 'experts' likely relevant to the topic, and later each word is sent to the best two in that shortlist. This double-step saves time, keeps the experts focused, and lets huge language models run faster and smarter.



DOI: 10.6084/m9.figshare.32952662

How we generate hypotheses for you:

1. Our system digests 100 million peer-reviewed papers to build the knowledge base.
2. A large language model then uses this knowledge base to generate 92,000 draft ideas, using about 30 million tokens of generation.
3. The drafts face 23,000 rapid AI match-ups to choose the top 50 ideas—a 0.05% acceptance rate.