DEPTHROUTER: PRIVACY-PRESERVING LLM INFERENCE VIA JOINT DEPTH-AXIS PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Secure multi-party computation (MPC) protects both the user's prompt and the provider's weights during large language model (LLM) inference, but makes every executed layer expensive. Reducing model depth could lower the cost of private LLM inference, but removing blocks can degrade answer quality, while input-dependent selection introduces additional privacy and computation requirements. **DepthRouter** addresses this design problem by constructing pruned model variants with LoRA-based quality recovery and studying query-level selection among them. Variants skip FFN and attention sub-blocks under public, static schedules chosen with an MPC cost model, and adapters placed next to the skipped sub-blocks are distilled from the unpruned model. On Llama-3.1-8B with MPCache, a static variant that skips 12 FFN sub-blocks retains 51.70 of 54.71 average F1 on LongBench QA at a 22.8% estimated reduction in MPC cost. Selection among variants shows clear headroom: an oracle gains $+4.7$ F1 on QA at 9% lower estimated cost and $+7.0$ accuracy points on TREC, while our learned quality estimator matches, but does not yet exceed, the best fixed variants.

1 INTRODUCTION

Large language models (LLMs) (Touvron et al., 2023; Grattafiori et al., 2024) are increasingly deployed as cloud-hosted Machine-Learning-as-a-Service (MLaaS), but this raises serious privacy concerns: users must upload prompts that may contain sensitive personal information, while service providers are unwilling to release proprietary model weights to the client. Secure Multi-Party Computation (MPC) (Goldreich et al., 1987; Mohassel & Zhang, 2017; Knott et al., 2021) addresses both sides of this problem by enabling client and server to jointly run inference over secret-shared inputs without revealing either query or weights, and has become a mainstream research direction for privacy-preserving LLM inference (Zeng et al., 2025; Dong et al., 2023; Hao et al., 2022; Lu et al., 2025).

However, MPC-based LLM inference is prohibitively slow. We profile a single Llama-3.1-8B decoder layer on SecretFlow SPU (Ma et al., 2023) (ABY3, 3PC, FM64) and find that each decode token takes minutes per layer—a 300–1100$\times$ wall-clock slowdown over plaintext (Figure 1, left). Per element, Softmax costs $174\times$, SiLU $53\times$, and RMSNorm $85\times$ more than matrix multiplication in our measurements (Table 4), and FFN and attention account for 50–80% and 20–50% of per-layer MPC time across contexts of 100 to 10,000 tokens. Existing work makes each layer cheaper, through protocol optimization (Mohassel & Rindal, 2018; Dong et al., 2023; Lu et al., 2025; Pang et al., 2024), MPC-friendly operator approximations (Li et al., 2023; Luo et al., 2024), fine-tuning-time model changes (Rathee et al., 2024), and KV-cache compression (Zeng et al., 2025). A complementary lever is to execute fewer layers: a sub-block that is not evaluated saves its entire MPC cost, however cheaply it would otherwise have been computed. This leads to our research question: *Can we reduce the depth of computation required for private LLM inference while preserving answer quality and accounting for the cost of private selection?*

Answering it requires addressing three challenges, each of which motivates one part of DepthRouter.

Challenge 1: removing blocks saves computation but can damage answers. On Llama-3.1-8B over four LongBench QA datasets, skipping just 2 FFN sub-blocks without recovery drops average F1 from 54.71 to 28.01, and skipping 6 drops it to 11.65 (Figure 2). This loss comes from removing computation, not from MPC; MPC only raises the reward for removing it. *Response: pruned variants*

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.

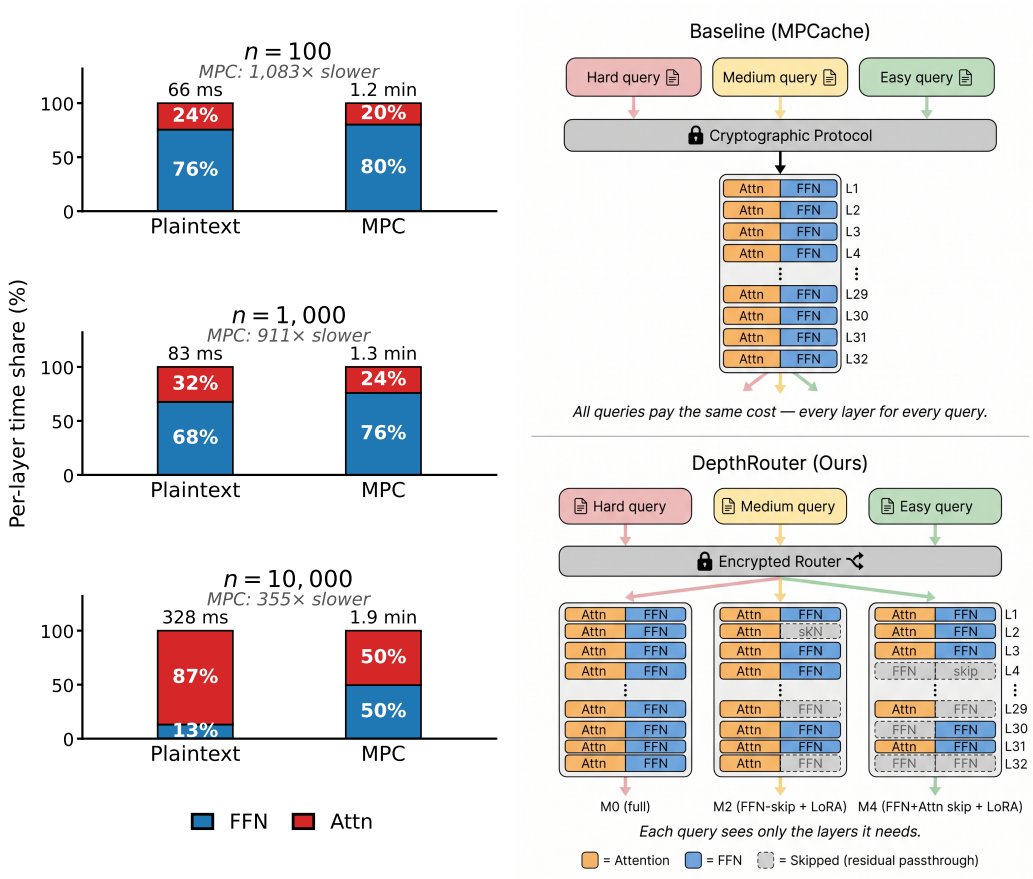


Figure 1: **Left:** per-layer cost of one Llama-3.1-8B decoder under plaintext vs. MPC at three context lengths—both FFN (50–80%) and attention (20–50%) are substantially inflated under MPC, exposing the depth axis as a high-leverage target. **Right:** DepthRouter's depth-aware router sends each query to the pruned-and-healed 8B variant it predicts to be the cheapest good-enough one, instead of paying full-model MPC cost for every query.

with LoRA-based distillation (Section 3.2). With the backbone frozen, we place rank-16 LoRA adapters on the layers adjacent to each skipped sub-block—FFN projections for FFN skips, attention projections for attention skips—where the representation shift caused by the skip must be absorbed. The adapters are trained by KL+CE distillation from the unpruned teacher with all of the variant's skips enabled, so they compensate for the combined change rather than for each skip in isolation. Skipping 7 FFN sub-blocks with this recovery retains 53.2 F1.

Challenge 2: private execution changes the cost of computation and of selection. Under MPC, the cost of a sub-block depends on the protocol and on context length rather than on plaintext FLOPs, and any decision that depends on the input must itself be computed securely. *Response: MPC cost estimates, static schedules, and fixed-capacity token selection (Sections 3.1, 3.3, 3.6).* We decide what to remove with a cost model built from measured per-operator MPC costs. A skipped FFN removes its three projections, SiLU, and pre-FFN RMSNorm; a skipped attention removes its projections, score computation, Softmax, and RMSNorm. Skip schedules are public and fixed, so they involve no secure selection. Token-level adaptivity is optional: fixed-capacity MoD gates (Raposo et al., 2024) keep tensor sizes public, but gate scoring, oblivious top-k, and gather/scatter remain as secure selection operations. Our savings figures estimate the computation removed by skipping (Section 3.6).

Challenge 3: a single pruning level applies the same trade-off to every query. Recovered variants make complementary errors: on LongBench QA, for at least 28% of queries a pruned variant scores higher than the full model, so a fixed choice of variant leaves quality or cost on the table. *Response: a quality estimator selects among variants (Section 3.5).* A BERT-mini estimator predicts which

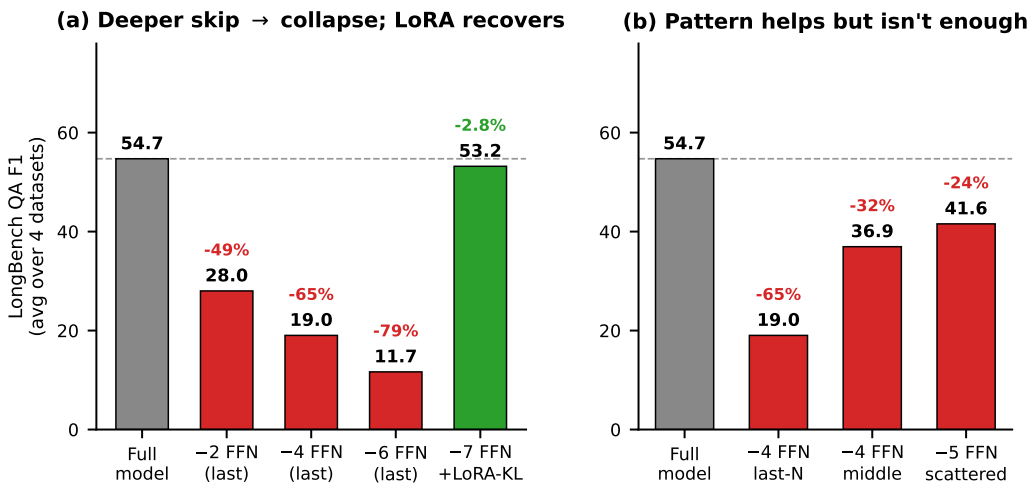


Figure 2: **Unhealed sub-block skipping degrades quality.** Average LongBench QA F1 on Llama-3.1-8B vs. skip depth (a) and skip pattern (b), evaluated in plaintext; the loss is caused by removing sub-blocks, not by MPC. LoRA-KL healing (green) recovers most of the lost quality.

variant is the cheapest one that retains at least 90% of the full model's quality on the query, and the query is sent to that variant. Selection happens once per query, before decoding, so execution follows one of K public static graphs, and the selected index is visible to the servers (Section 2.2). Whether learned selection beats a fixed variant is an empirical question we evaluate separately (Section 4.3).

Positioning. DepthRouter studies depth reduction for private LLM inference by combining quality recovery after pruning with MPC cost accounting and selection among model variants. Layer pruning (Men et al., 2024) and post-pruning healing (Gromov et al., 2025), LoRA distillation (Hu et al., 2022; Hinton et al., 2015), fixed-capacity MoD (Raposo et al., 2024), and quality-based routing (Chen et al., 2024; Ong et al., 2025; Dekoninck et al., 2025) are established ingredients. What DepthRouter adds is their adaptation to private inference: FFN and attention sub-blocks are scheduled separately using measured MPC operator costs instead of layer counts; recovery adapters are placed around the skipped sub-blocks and trained under each variant's full skip configuration; and the selection targets are depth variants of one backbone with public schedules, selected once per query.

Our contributions are:

- **A depth-reduction design.** We construct pruned variants with public FFN and attention skip schedules and recover answer quality through targeted LoRA distillation (Section 3). The static FFN-skip configuration that skips 12 FFN sub-blocks retains 51.70 vs. 54.71 F1 on LongBench QA at a 22.8% estimated reduction in MPC cost (Table 2).
- **A cost–quality evaluation.** We quantify these trade-offs under an MPC cost model built from measured per-operator SPU costs (Section 3.6), reporting savings as model estimates distinct from measured end-to-end execution time.
- **A query-selection study.** We evaluate learned selection among recovered variants against fixed configurations and an oracle (Section 4.3). Oracle selection gains $+4.7$ F1 at 9% lower estimated cost on QA and $+7.0$ accuracy points on TREC, whereas the current learned estimator matches but does not exceed the best fixed configurations, identifying selection accuracy as the main remaining limitation.

2 Background and Related Work

2.1 MPC for Neural Network Inference

Secure Multi-Party Computation enables N parties to jointly compute a function $f(x_1, \ldots, x_N)$ without revealing individual inputs (Goldreich et al., 1987; Yao, 1986). For LLM inference under the semi-honest (honest-but-curious) model (Evans et al., 2018), the client secret-shares its query among non-colluding computing parties, which execute the model on additive secret shares over $\mathbb{Z}_L$; our measurements use the three-party ABY3 protocol (Mohassel & Rindal, 2018) in SPU. Additions and multiplications by public constants are local; multiplications between secret-shared values, including secret weights, require interaction, and nonlinear operations (Softmax, GELU/SiLU, RMSNorm) require multi-round protocols and are more expensive still (Knott et al., 2021; Hao et al., 2022; Dong et al., 2023).

The per-element MPC operation costs underlying the multipliers cited in Section 1 (Softmax 1004.7 μs/elem, RMSNorm 493.3 μs/elem, SiLU 306.8 μs/elem, MatMul 5.77 μs/elem) are measured on SecretFlow SPU and listed in Table 4 (Appendix A).

2.2 Threat Model

DepthRouter operates under the semi-honest (or honest-but-curious) adversary model that is standard in privacy-preserving deep learning (Evans et al., 2018; Goldreich et al., 1987). Every participant—the client issuing the query and the non-colluding servers executing the secret-shared computation—faithfully follows the protocol, but may attempt to learn more than its prescribed output by inspecting received messages, intermediate share values, and any memory transcript exposed during execution. We do not model malicious deviation from the protocol, model-extraction attacks, or external side channels such as timing, cache, or memory-access patterns; these exclusions match the convention adopted by prior MPC LLM inference systems (Zeng et al., 2025; Dong et al., 2023; Li et al., 2023).

Within this model, both the variant-selection router and the candidate model pool execute on the server side; the client only ever transmits its query as additive secret shares, so no single server reconstructs the plaintext. The pool's architectural metadata—skip schedules, MoD capacities, and LoRA adapter placement—is treated as public, in line with the public-architecture / private-weights convention used in earlier MPC LLM frameworks (Li et al., 2023; Zeng et al., 2025).

Routing leakage and building blocks. The router reveals only the selected index $m^*(x)$, which is part of the ideal functionality ($\lceil \log_2 K \rceil$ bits per query); the servers then run the fixed public graph of $M_{m^*(x)}$. Variants, router, and optional MoD gates use existing MPC operations (secret-shared linear layers, nonlinear approximations, secure comparison, oblivious top-k), and their security follows by sequential composition with the routing leakage as the only addition. Fixing one public variant removes this leakage.

2.3 Related Work

Token-axis compression for MPC LLM inference. MPCache (Zeng et al., 2025) evicts low-importance KV-cache entries on the token axis, combining static prefill-phase eviction with dynamic decode-phase hierarchical clustering; CipherPrune (Zhang et al., 2025) similarly prunes tokens during cryptographic private Transformer inference. DepthRouter is orthogonal: it reduces depth-axis cost on billion-parameter LLMs and composes with MPCache as the attention backbone.

Architectural optimization for MPC inference. MPCFormer (Li et al., 2023) replaces MPC-unfriendly operators and distills; Marill (Rathee et al., 2024) applies layer freezing, low-rank adaptation, and head merging. DepthRouter is likewise architectural, but removes sub-blocks of a pretrained dense model and routes each query among several recovered variants.

Depth reduction in plaintext. Three families have been explored: *layer pruning* (Men et al., 2024; Gromov et al., 2025; Lu et al., 2024; Fan et al., 2020), including low-rank healing after layer removal (Gromov et al., 2025); input-dependent sparsity and layer skipping (Liu et al., 2023; Yang et al., 2025); *early exit* (Teerapittayanon et al., 2016; Schuster et al., 2022; Elhoushi et al., 2024; Xin et al., 2020; Del Corro et al., 2023); and *Mixture of Depths* (Raposo et al., 2024), which routes

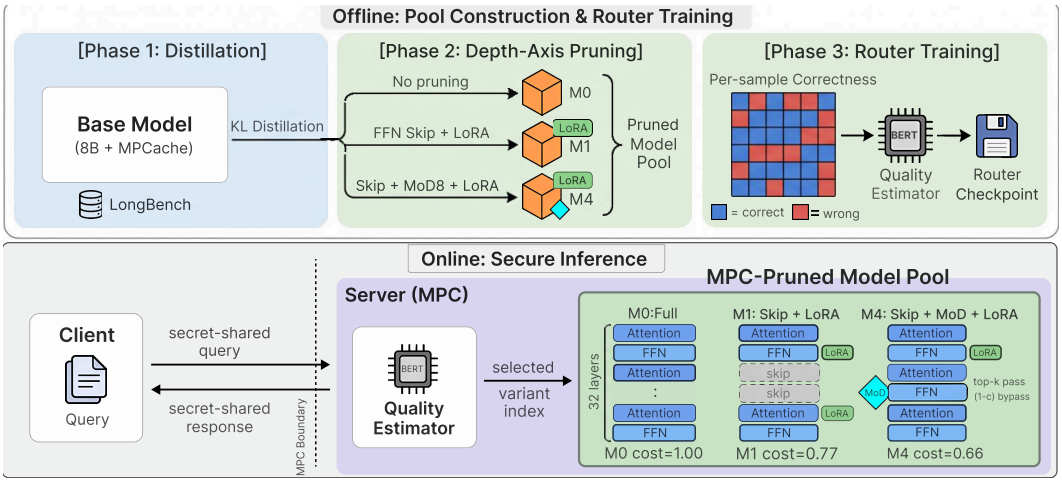


Figure 3: **DepthRouter pipeline.** Offline (top): sub-blocks are removed under public static schedules, LoRA-KL distillation recovers each pruned variant, and a quality estimator is trained to predict the cheapest good-enough variant. Online (bottom): the estimator selects one variant per query; the selected index is visible to the servers, and the chosen variant's fixed graph is executed under MPC. Costs are model estimates.

tokens through blocks with a fixed top-k capacity. AdaSkip (He et al., 2025) is a strong training-free plaintext sub-layer skipper.

LLM routing. FrugalGPT (Chen et al., 2024), RouteLLM (Ong et al., 2025), and cascade routing (Dekoninck et al., 2025) route queries among independently trained LLMs in plaintext, and Wu et al. (2026) execute such a router under MPC with MPC-friendly encoder operations and secure top-k. DepthRouter instead co-designs the routing targets: it routes among depth variants of one backbone that differ only in public skip schedules and small adapters, with granularity and leakage chosen for MPC; MPC-friendly router designs are complementary.

Concurrent MPC-side routing. CryptoMoE (Zhou et al., 2025) privatizes expert routing in sparse MoE backbones with dedicated secure dispatch/combine protocols. DepthRouter targets dense LLMs, routes over depth variants per query, and needs no new dispatch protocol.

3 Method

DepthRouter operates on a frozen Llama-3.1-8B-Instruct backbone with MPCache attention and changes which sub-blocks are evaluated, not how any operator is evaluated under MPC: Sections 3.1–3.4 build the variant pool and Section 3.5 routes among it. Figure 3 illustrates the offline pool-construction stage and the online per-query variant selection.

3.1 MPC Co-Design Principle: FFN–Attention Asymmetry

Many plaintext depth-reduction methods remove or skip whole layers (Men et al., 2024; Elhoushi et al., 2024), but under MPC the two sub-blocks differ in cost and quality sensitivity: FFN linear layers dominate per-layer MPC cost at moderate context (Table 1) while attention's share grows with context, and in our experiments attention removal hurts quality more, whereas FFN tolerates substantial removal once paired with LoRA healing. DepthRouter therefore makes FFN the primary skip target (Section 3.2); mid-late attention sub-blocks are removed only as a complementary lever (Section 3.4), with the boundary set $\{0, 1, 2, 30, 31\}$ preserved across all variants.

3.2 Static FFN-Layer Skipping with KL-Distillation LoRA Healing

Zero-shot static skip collapses past $\sim$5 scattered FFN sub-blocks (Figure 2). To push past this quality ceiling, we install rank-r LoRA adapters (Hu et al., 2022) on FFN projections (`gate_proj`,

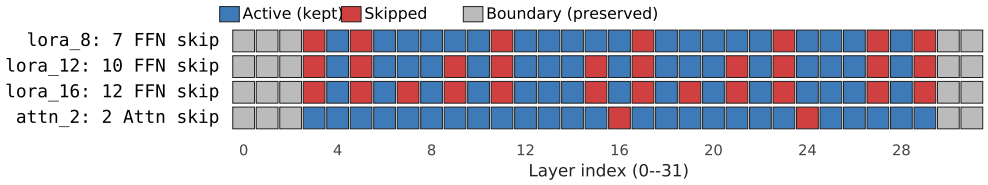


Figure 4: **Skip-pattern bitmap for the four schedules.** Each row is one schedule across the 32 Llama-3.1-8B sub-blocks; cells show whether the FFN (rows 1–3) or attention (row 4) at that depth is kept (blue), skipped (red), or in the always-preserved boundary set $\{0, 1, 2, 30, 31\}$ (grey).

`up_proj`, `down_proj`) of the layers neighboring the skipped set; the backbone is frozen and only LoRA parameters $A \in \mathbb{R}^{r\times d}$, $B \in \mathbb{R}^{d'\times r}$ are trained. We use the scattered uniform-K family as the underlying skip pattern (the strongest zero-shot family in Figure 2); as a design choice, the boundary set $\{0, 1, 2, 30, 31\}$ is never skipped, and the concrete schedules (`lora_8`, `lora_12`, `lora_16`) are shown in Figure 4 and listed in Appendix A.2.

Training objective. We use combined KL+CE distillation with an inline teacher:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{CE}} + (1-\alpha) \cdot T^2 \cdot \text{KL}\Big(p^{(T)}_{\text{teacher}} \parallel p^{(T)}_{\text{student}}\Big) \tag{1}$$

where the teacher is the same model with FFN-skip disabled and LoRA scale set to zero (no_grad forward), and $p^{(T)}$ denotes temperature-scaled softmax with $\alpha = 0.2$, $T = 2.0$. The loss is computed only on answer tokens; the KL term matches the teacher's full output distribution.

3.3 Mixture of Depths under MPC

Following MoD (Raposo et al., 2024), we use a fixed-capacity gate at each MoD layer l: a linear gate $g_l : \mathbb{R}^d \to \mathbb{R}$ scores each token's hidden state and the top-k tokens pass through the FFN, with the remaining $(1-c)\cdot n$ tokens skipping via residual ($c = k/n$ fixed and public). Fixed capacity is inherited from MoD; under MPC the identities of selected tokens must also stay hidden, so top-k selection and gather/scatter use standard oblivious building blocks (e.g., a Batcher bitonic sorting network (Batcher, 1968)), not a new protocol, and their overhead counts against the FFN savings. MoD gates and LoRA adapters are trained jointly under the same KL+CE objective (Eq. 1); the teacher forward disables both, the student enables both.

3.4 Attention-Layer Skipping (Complementary Lever)

We dually monkey-patch `LlamaAttention.forward` to short-circuit marked layers ($h_{\text{out}} = h_{\text{in}}$). Motivated by the observation that many deeper layers can be removed with limited loss (Gromov et al., 2025), we keep the boundary set $\{0, 1, 2, 30, 31\}$ and target scattered mid-late indices; attention-side LoRA on `q/k/v/o_proj` of neighboring layers heals further, trained jointly with FFN-side adapters under the same KL+CE objective (Eq. 1). Per skipped attention layer the savings are $\sim 1\%$ of total MPC time at n=100 rising to $\sim 1.6\%$ at n=10,000 (attention is 20–50% of layer time, with its share growing with context). Attention skipping is less cost-efficient per-F1 than FFN-skip at moderate n but provides an orthogonal knob for the router and scales better at long context.

3.5 Quality Estimator for Variant Selection

Goal. Sections 3.2–3.4 produce K progressively pruned variants of the same 8B backbone, $\mathcal{M} = \{M_0, \ldots, M_{K-1}\}$, with increasing estimated MPC savings and decreasing average quality. Selection sends each query to the cheapest variant that is still *good enough* for that input, with M_0 (the full model) as the fallback.

Labels and estimator. For a query x with per-sample score $s(M_i, x)$ (token F1 for QA, exact match for TREC), the label is the cheapest variant that retains at least 90% of the full model's score,

$$y(x) = \arg\min_{0 \le i < K} \text{cost}(M_i) \quad \text{s.t.} \quad s(M_i, x) \ge 0.9\, s(M_0, x), \tag{2}$$

which always exists because M_0 satisfies the condition; when $s(M_0, x){=}0$, the cheapest variant is chosen. The estimator is a BERT-mini encoder ($\sim$11M parameters) with a K-way classification head, trained with cross-entropy to predict $y(x)$; at inference the query is sent to the predicted variant $m^*(x)$. This is single-step selection before execution, without the sequential cascading of Dekoninck et al. (2025). The estimator is evaluated with 5-fold cross-validation, so each query is scored by a model that did not train on it (details in Appendix A.3). For the TREC pool, the estimator instead has K sigmoid quality heads $q_i(x)$, trained with binary cross-entropy on the per-variant labels $\mathbf{1}[s(M_i, x) \ge 0.9\, s(M_0, x)]$; the query is sent to the cheapest variant with $q_i(x) \ge \tau$, falling back to M_0, and Table 3 sweeps τ.

3.6 MPC Cost Model

We compute per-layer MPC time using the standard transformer cost decomposition (QKV projection, attention scores/output, Softmax, SwiGLU FFN, two RMSNorms), with each component multiplied by its per-element MPC cost from Table 4. For Llama-3.1-8B at $n{=}4{,}000$ context, this yields the breakdown in Table 1.

Table 1: Per-layer MPC cost decomposition for Llama-3.1-8B (1 decode token, $n{=}4{,}000$).

Component	Time (s)	% of layer	Nonlinear amplification
Attention (linear)	431.0	27.2%	1×
Attention (Softmax)	128.6	8.1%	174× per element
FFN (linear)	1016.3	64.1%	1×
FFN (SiLU)	4.4	0.3%	53× per element
RMSNorm (×2)	4.0	0.3%	85× per element
Total	**1584.3**	**100%**	

What the estimates include. Per-operator costs are measured on SPU; per-variant savings are model estimates, not measured end-to-end execution times. The saving of a variant is the summed cost of the components it removes, divided by the full model's per-decode-token cost at $n{=}4{,}000$ (32 decoder layers plus the LM head; Appendix B). A skipped FFN sub-block removes its three projections, SiLU, and pre-FFN RMSNorm ($\sim$1020 s per decode token); a skipped attention sub-block removes its projections, score computation, Softmax, and RMSNorm. For MoD we count only the FFN computation of the $(1{-}c)$ bypassed tokens. These figures cover the skipped computation; MoD selection operations (gate, oblivious top-k, gather/scatter) and the quality estimator are not included.

4 Experiments

4.1 Setup

Model. Llama-3.1-8B-Instruct with MPCache attention (KV-cache eviction).

Datasets. LongBench (Bai et al., 2024): HotpotQA (multi-hop QA), NarrativeQA (narrative QA), TriviaQA (few-shot QA), Qasper (scientific QA), and TREC (question-type classification). QA datasets use 200 samples each (800 total, metric: token-level F1); TREC uses 200 samples (metric: classification accuracy).

Training. Recovery adapters are self-distilled: the targets are the full model's own greedy outputs rather than gold answers, so no reference answers are used in training. Distillation uses the LongBench QA prompts with a random 90/10 train/validation split, and QA F1 is reported on the same 800 prompts. Distillation uses teacher-forced KL+CE ($\alpha{=}0.2$, $T{=}2.0$), AdamW (lr=2e-4), max context

Table 2: Non-dominated configurations on LongBench QA (4 datasets, 800 samples). Est. MPC saved is computed with the cost model of Section 3.6. ΔF1 relative to baseline (54.71).

Configuration	Method	Est. MPC Saved	Avg F1	ΔF1
Baseline (8B + MPCache)	—	0%	54.71	—
MoD_5+LoRA (c=0.5)	MoD+LoRA	4.7%	54.49	−0.22
lora_8 (KL)	Static+LoRA	13.3%	53.20	−1.51
MoD_{10}+LoRA (c=0.25)	MoD+LoRA	14.2%	52.98	−1.73
lora_16 r16 (KL+CE)	Static+LoRA	22.8%	51.70	−3.01

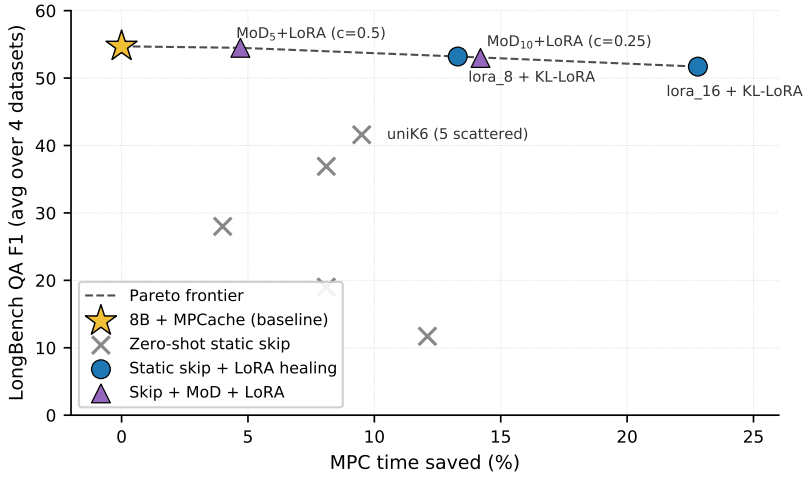


Figure 5: **LongBench QA Pareto frontier.** Zero-shot static skip (grey) collapses below 42 F1 even at modest MPC savings; LoRA healing (blue) and MoD+LoRA (purple) push the frontier (dashed) to within −3 F1 at 22.8% estimated savings.

4096 tokens, and gradient checkpointing, with MPCache disabled during training. Adapter placement, skip schedules, and parameter counts are in Appendix A.2.

Hardware. NVIDIA B200 GPUs (192 GB). MPC costs profiled on SecretFlow SPU v0.9.5 (ABY3, 3PC, FM64). End-to-end training on the joint FFN+attention LoRA on 800 samples completes in ∼28 minutes on a single B200 ($<$ 13M trainable parameters); the BERT-mini quality estimator trains in $<$ 1 GPU-hour.

Baselines. We compare against depth-reduction baselines without recovery. (i) *Zero-shot static skip* simply identity-skips a fixed subset of layers (Men et al., 2024; Gromov et al., 2025; Fan et al., 2020); we evaluate three pattern families—last-N, contiguous-middle, and scattered uniform-K (uniK-N). The strongest zero-shot configuration, `uniK6` (skip $\{5, 11, 17, 23, 29\}$), retains only 76% of baseline F1 at 9.5% MPC saved (Figure 2); pushing past 5 scattered FFN layers drops F1 below 5, and last-N already loses −26.7 F1 at just 2 skips. (ii) *Per-token Mixture-of-Depths* (Raposo et al., 2024) selects tokens as a function of secret-shared activations, so under MPC the selection must be hidden by oblivious selection and data movement, whose overhead offsets part of the savings. These motivate our jointly healed static architecture (§3.2) and per-query routing as the primary adaptive mechanism, with fixed-capacity MoD as an extension (§3.3).

4.2 Pool Construction: From Individual Methods to Pruned Variants

Table 2 lists the non-dominated configurations on LongBench QA. The zero-shot baseline (best point: scattered uniK6, 76% of baseline F1 at only 9.5% MPC saved; full curve in Figure 2) is dominated everywhere and does not appear in the table; the listed configurations are LoRA-healed static schedules and MoD with LoRA, the most aggressive being the static `lora_16` schedule (12 FFN sub-blocks, 22.8% estimated savings, −3.01 F1).

Table 3: DepthRouter results on TREC classification (200 samples, 5-fold CV). **Pool members** are static configurations; **DepthRouter** uses a learned quality-estimator router to select per query. Oracle = perfect routing (cheapest correct model per query). MPC cost covers the executed variant and excludes the router's own execution.

Method	TREC Acc	ΔAcc	Est. Cost	Est. Speedup
Baseline				
8B + MPCache	72.50	—	1.000	1.00×
Pool members (static schedule, no routing)				
M1: FFN-skip + LoRA	72.50	+0.00	0.772	1.30×
M2: + MoD_4@50%	71.50	−1.00	0.734	1.36×
M3: + MoD_8@50%	70.00	−2.50	0.696	1.44×
M4: + MoD_8@25%	66.50	−6.00	0.658	1.52×
DepthRouter (quality-estimator routing, τ sweep)				
$\tau = 0.8$	72.00	−0.50	0.872	1.15×
$\tau = 0.7$	70.50	−2.00	0.706	1.42×
$\tau = 0.5$	67.50	−5.00	0.672	1.49×
Upper bounds				
Oracle (perfect router)	79.50	+7.00	0.742	1.35×

Cost of recovery. Recovery for `lora_16 r16` trains <13M LoRA parameters in ∼28 minutes on a single B200 (Appendix A.2); the estimated saving of ∼1020 s of MPC time per skipped FFN sub-block per decode token recurs for every query.

4.3 Query Selection among Variants

We compare learned selection with every fixed variant and with an oracle that applies the label rule of Eq. 2 to the true scores; costs are estimates (Section 3.6).

QA pool. M_0 is the full model (cost 1.000), M_1 skips attention $\{16, 24\}$ with LoRA (0.979), and M_2 additionally skips FFN $\{5, 11, 17, 23, 29\}$ with joint LoRA (0.884). This experiment scores TriviaQA answers without truncating them to the first line, so its full-model baseline is 36.35 F1 rather than the 54.71 of Table 2. Fixed variants reach 36.35, 36.49, and 34.76 F1. The learned estimator reaches 34.80 F1 at cost 0.885, essentially the same as always choosing M_2; the oracle reaches 41.02 F1 at cost 0.910, +4.67 F1 over the full model.

TREC pool. We stack MoD on the `lora_16` backbone to obtain a five-member pool covering 0%–34.2% estimated savings (Table 3).

Analysis. M1 matches the baseline (72.50) at 22.8% lower estimated cost. The oracle reaches 79.5%, 7 points above any fixed variant, because variants err on different queries. The learned estimator does not yet exploit this: the τ=0.8 point (72.0% at 0.872) is dominated by M1, τ=0.7 (70.5% at 0.706) lies on the frontier between M3 and M2, and τ=0.5 lies below it. On both pools, selection therefore matches but does not exceed the best fixed variants; closing the oracle gap is the main open problem (Appendix C).

5 Conclusion

We asked whether the depth of computation in private LLM inference can be reduced while preserving answer quality and accounting for the cost of private selection. Our results indicate that static sub-block skipping with targeted LoRA distillation retains most of the answer quality at a substantial estimated MPC saving, and that recovered variants make complementary errors, giving query-level selection clear headroom. Improving the learned estimator toward the oracle, and measuring secure selection and end-to-end MPC execution, are natural next steps (Appendix C).

AI USE STATEMENT

Generative AI tools were used to assist with figure creation and with preparing the manuscript in the ICLR submission format. The authors take responsibility for the final content of this work, including all AI-assisted figures and text.

ETHICS STATEMENT

DepthRouter aims to reduce the computational cost of privacy-preserving LLM inference. Its privacy guarantees are limited to the semi-honest threat model described in Section 2.2, under which the per-query router reveals the selected variant index; they do not address malicious participants or external side channels. Computational privacy does not remove the biases or potential harmful outputs of the underlying language model.

REPRODUCIBILITY STATEMENT

Section 3 describes the pruning, healing, and routing procedures. Appendix A documents the software environment, MPC profiling setup, and router hyperparameters; Appendix B gives the analytical cost decomposition used to estimate MPC savings. The experimental tables report the evaluated configurations and their quality–cost tradeoffs.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Kenneth E. Batcher. Sorting Networks and Their Applications. In *Proceedings of the AFIPS Spring Joint Computer Conference*, pp. 307–314, 1968.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Transactions on Machine Learning Research*, 2024.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A Unified Approach to Routing and Cascading for LLMs. In *International Conference on Machine Learning (ICML)*, 2025.

Luciano Del Corro, Allie Del Giorno, Sahaj Agarwal, Bin Yu, Ahmed Awadallah, and Subhabrata Mukherjee. SkipDecode: Autoregressive Skip Decoding with Batching and Caching for Efficient LLM Inference. *arXiv preprint arXiv:2307.02628*, 2023.

Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Chen. PUMA: Secure Inference of LLaMA-7B in Five Minutes. *arXiv preprint arXiv:2307.12533*, 2023.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642, 2024. doi: 10.18653/v1/2024.acl-long.681.

David Evans, Vladimir Kolesnikov, and Mike Rosulek. A Pragmatic Introduction to Secure Multi-Party Computation. *Foundations and Trends in Privacy and Security*, 2(2-3):70–246, 2018.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing Transformer Depth on Demand with Structured Dropout. In *International Conference on Learning Representations (ICLR)*, 2020.

Oded Goldreich, Silvio Micali, and Avi Wigderson. How to Play Any Mental Game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pp. 218–229, 1987.

Aaron Grattafiori et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The Unreasonable Ineffectiveness of the Deeper Layers. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=ngmEcEer8a.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private Inference on Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/64e2449d74f84e5b1a5c96ba7b3d308e-Abstract.html.

Zhuomin He, Yizhen Yao, Pengfei Zuo, Bin Gao, Qinya Li, Zhenzhe Zheng, and Fan Wu. AdaSkip: Adaptive Sublayer Skipping for Accelerating Long-Context LLM Inference. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. CrypTen: Secure Multi-Party Computation Meets Machine Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P. Xing, and Hao Zhang. MPCFormer: Fast, Performant and Private Transformer Inference with MPC. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=CWmvjOEhgH-.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Ré, and Beidi Chen. Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time. In *International Conference on Machine Learning (ICML)*, 2023.

Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and Wenguang Chen. BumbleBee: Secure Two-party Inference Framework for Large Transformers. In *Network and Distributed System Security Symposium (NDSS)*, 2025.

Yao Lu, Hao Cheng, Yujie Fang, Zeyu Wang, Jiaheng Wei, Dongwei Xu, Qi Xuan, Xiaoniu Yang, and Zhaowei Zhu. Reassessing Layer Pruning in LLMs: New Insights and Methods. *arXiv preprint arXiv:2411.15558*, 2024.

Jinglong Luo, Yehong Zhang, Zhuo Zhang, Jiaqi Zhang, Xin Mu, Hui Wang, Yue Yu, and Zenglin Xu. SecFormer: Fast and Accurate Privacy-Preserving Inference for Transformer Models via SMPC. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-acl.790.

Junming Ma, Yancheng Zheng, Jun Feng, Derun Zhao, Haoqi Wu, Wenjing Fang, Jin Tan, Chaofan Yu, Benyu Zhang, and Lei Wang. SecretFlow-SPU: A Performant and User-Friendly Framework for Privacy-Preserving Machine Learning. In *USENIX Annual Technical Conference (ATC)*, 2023.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. *arXiv preprint arXiv:2403.03853*, 2024.

Payman Mohassel and Peter Rindal. ABY3: A Mixed Protocol Framework for Machine Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 35–52, 2018.

Payman Mohassel and Yupeng Zhang. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 19–38. IEEE, 2017.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs from Preference Data. In *International Conference on Learning Representations (ICLR)*, 2025.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. BOLT: Privacy-Preserving, Accurate and Efficient Inference for Transformers. In *IEEE Symposium on Security and Privacy (S&P)*, 2024.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-Depths: Dynamically Allocating Compute in Transformer-Based Language Models. *arXiv preprint arXiv:2404.02258*, 2024.

Deevashwer Rathee, Dacheng Li, Ion Stoica, Hao Zhang, and Raluca Popa. MPC-Minimized Secure LLM Inference. *arXiv preprint arXiv:2408.03561*, 2024.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q Tran, Yi Tay, and Donald Metzler. Confident Adaptive Language Modeling. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks. In *23rd International Conference on Pattern Recognition (ICPR)*, 2016.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023.

Xidong Wu, Yukuan Zhang, Yuqiong Ji, Reza Shirkavand, Qian Lou, and Shangqian Gao. Privacy-Preserving LLMs Routing. *arXiv preprint arXiv:2604.15728*, 2026.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference. In *Proceedings of ACL*, pp. 2246–2251, 2020.

Ning Yang, Fangxin Liu, Junjie Wang, Tao Yang, Kan Liu, Haibing Guan, and Li Jiang. DASH: Input-Aware Dynamic Layer Skipping for Efficient LLM Inference with Markov Decision Policies. *arXiv preprint arXiv:2505.17420*, 2025.

Andrew Chi-Chih Yao. How to Generate and Exchange Secrets. In *27th Annual Symposium on Foundations of Computer Science*, pp. 162–167, 1986.

Wenxuan Zeng, Ye Dong, Jinjin Zhou, Jin Tan, Lei Wang, Tao Wei, Runsheng Wang, and Meng Li. MPCache: MPC-Friendly KV Cache Eviction for Efficient Private LLM Inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. CipherPrune: Efficient and Scalable Private Transformer Inference. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/83a14a36de4502bac5b580db36e81858-Abstract-Conference.html.

Yifan Zhou, Tianshi Xu, Jue Hong, Ye Wu, and Meng Li. CryptoMoE: Privacy-Preserving and Scalable Mixture of Experts Inference via Balanced Expert Routing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2511.01197.

A Reproducibility Details

A.1 Software and Environment

- **Python** 3.10, **PyTorch** 2.1, `transformers` 4.40, `datasets` 2.18, `accelerate` 0.27.
- **Backbone:** Llama-3.1-8B-Instruct (`unsloth/Meta-Llama-3.1-8B-Instruct`), bfloat16, eager attention.
- **KV-cache attention:** MPCache (Zeng et al., 2025) reference implementation; static eviction at 40% retention by default, dynamic selection ratios $0.5/0.2$ as in the original release.
- **MPC profiling:** SecretFlow SPU v0.9.5 with the ABY3 backend, 3-party setting, fixed-point ring $\mathbb{Z}_{2^{64}}$ (FM64). All per-element costs in Table 4 are wall-clock medians over 100 invocations on a quiet machine.
- **Hardware:** NVIDIA B200 (192 GB) GPUs; one GPU per training/inference job.

Table 4: Per-element MPC operation costs (SecretFlow SPU, ABY3/FM64), referenced in Section 2.

Operation	Cost (μs/elem)	vs. MatMul	Where used
MatMul	5.77	$1\times$	QKV, Attn, FFN
SiLU	306.78	$53\times$	FFN gate
RMSNorm	493.32	$85\times$	Pre-attn, pre-FFN
Softmax	1004.70	$174\times$	Attention

A.2 Joint LoRA Distillation Hyperparameters

- **Adapter placement.** For an FFN-skip schedule S_{ffn}, LoRA adapters are installed on FFN projections (`gate_proj`, `up_proj`, `down_proj`) of every layer in $\textsc{Neighbors}(S_{\text{ffn}})$ (immediate neighbors clipped to $[0, L-1]$, excluding skipped layers). For an attention-skip schedule S_{attn}, attention-side adapters are installed on `q_proj`/`k_proj`/`v_proj`/`o_proj` of $\textsc{Neighbors}(S_{\text{attn}})$.
- **Rank/scale.** Rank $r = 16$, $\alpha = 16$ (so the effective LoRA scale equals $\alpha/r = 1$). FFN skip schedules: `lora_8` $= \{3, 5, 11, 17, 23, 27, 29\}$ (7 FFN), `lora_12` $= \{3, 5, 9, 11, 15, 17, 21, 23, 27, 29\}$ (10 FFN), `lora_16` $= \{3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 27, 29\}$ (12 FFN); attention schedule `attn_2` $= \{16, 24\}$. The boundary set $\{0, 1, 2, 30, 31\}$ is never skipped across any schedule. Figure 4 (Section 3.2) visualises which sub-blocks are kept versus skipped per schedule. Trainable parameter counts: `lora_16` (rank-16 on FFN `gate_proj`/`up_proj`/`down_proj` of $\sim$15 neighbor layers) $\approx$13M; `attn_2` + LoRA only $=$1,703,936.
- **Distillation loss.** Combined KL + cross-entropy on answer tokens only: $\mathcal{L} = \alpha\mathcal{L}_{\text{CE}} + (1-\alpha)\,T^2\,\text{KL}(p^{(T)}_{\text{teacher}} \| p^{(T)}_{\text{student}})$, with $\alpha = 0.2$, $T = 2.0$. The teacher is the same model with FFN/attn skips disabled and LoRA scale set to 0 (`no_grad`).
- **Optimizer.** AdamW, learning rate 2×10^{-4}, weight decay 0.0. Batch size 1, gradient accumulation 4 (effective batch 4). 2 epochs over ~ 720 training pairs and ~ 80 validation pairs (90/10 split of the per-sample teacher-forced LongBench-QA labels).
- **Memory.** Gradient checkpointing enabled; `model.enable_input_require_grads()` so the embedding output participates in the backward graph.

A.3 Pool-Router Hyperparameters

- **Encoder.** BERT-mini (`prajjwal1/bert-mini`, 4 layers, hidden 256, 11.3M parameters).
- **Classification head.** A linear projection $\mathbb{R}^{256} \to \mathbb{R}^K$ on the `[CLS]` token, trained with cross-entropy; the query is sent to the $\arg\max$ variant.
- **TREC quality heads.** For the TREC pool, K linear heads $\mathbb{R}^{256} \to \mathbb{R}$ with sigmoids predict $q_i(x)$, trained with binary cross-entropy on $\mathbf{1}[s(M_i, x) \geq 0.9 \cdot s(M_0, x)]$; the query goes to the cheapest variant with $q_i(x) \geq \tau$, or to M_0 if none qualifies.
- **Labels.** For each query, the label is the cheapest variant M_i whose per-sample score (token F1, or exact match for TREC) satisfies $s(M_i, x) \geq 0.9 \cdot s(M_0, x)$ (Eq. 2); M_0 always qualifies. If $s(M_0, x) = 0$ the condition holds for every variant and the cheapest variant is the label. On the QA pool this yields 600 / 118 / 82 queries labelled M_2 / M_1 / M_0.

- **Oracle.** The QA oracle sends each query to its label and reports the realized score of that variant; it is not the best-scoring variant per query.
- **Optimization.** AdamW, lr 5×10^{-5}, batch size 32, 20 epochs per fold, max sequence length 128. 5-fold stratified CV (sklearn, seed 42); held-out predictions are aggregated across folds for the final F1/cost summary. The best epoch is selected by accuracy on each fold's held-out split.

B MPC Cost Model: FLOP Decomposition

For Llama-3.1-8B with $L = 32$ layers, hidden $d = 4096$, FFN inner $d_{\text{ff}} = 14336$, attention heads $H = 32$, KV heads $H_{\text{kv}} = 8$, and head dimension $d_h = d/H = 128$, per-decode-token per-layer costs decompose as:

Attention.

$$C_{\text{attn}}^{\text{linear}} = \underbrace{d \cdot (H + 2H_{\text{kv}}) \cdot d_h}_{\text{QKV proj}} + \underbrace{H \cdot n \cdot d_h}_{\text{score logits}} + \underbrace{H \cdot n \cdot d_h}_{\text{value gather}} + \underbrace{d^2}_{\text{out proj}} \tag{3}$$

$$C_{\text{attn}}^{\text{softmax}} = H \cdot n \cdot c_{\text{softmax}} \tag{4}$$

FFN (SwiGLU).

$$C_{\text{ffn}}^{\text{linear}} = 3 \cdot d \cdot d_{\text{ff}} \cdot c_{\text{matmul}} \quad \text{(gate + up + down)} \tag{5}$$

$$C_{\text{ffn}}^{\text{silu}} = d_{\text{ff}} \cdot c_{\text{silu}} \tag{6}$$

RMSNorm. $C_{\text{norm}} = d \cdot c_{\text{RMSNorm}}$, applied twice per layer (pre-attention, pre-FFN).

Each $c_{(\cdot)}$ is the per-element MPC cost in Table 4. Table 1 reports the resulting decomposition for $n = 4{,}000$.

Savings formulas. For a static FFN-skip schedule of $|S_{\text{ffn}}| = S$ sub-blocks:

$$\Delta T_{\text{static-ffn}} = S \cdot \left(C_{\text{ffn}}^{\text{linear}} + C_{\text{ffn}}^{\text{silu}} + C_{\text{norm}}\right). \tag{7}$$

For static attention-skip of A sub-blocks (with neighbour LoRA optionally healing nearby attention layers):

$$\Delta T_{\text{static-attn}} = A \cdot \left(C_{\text{attn}}^{\text{linear}} + C_{\text{attn}}^{\text{softmax}} + C_{\text{norm}}\right). \tag{8}$$

For Mixture-of-Depths with capacity c on M FFN layers (gate runs always; only the top-k tokens go through the FFN body):

$$\Delta T_{\text{MoD}} = M \cdot (1 - c) \cdot \left(C_{\text{ffn}}^{\text{linear}} + C_{\text{ffn}}^{\text{silu}}\right). \tag{9}$$

The pre-FFN RMSNorm and the gate linear ($d \to 1$, negligible) are still executed. The percent-MPC-saved figures throughout the paper are computed as $\sum \Delta T / T_{\text{full}}$ where $T_{\text{full}} = L \cdot T_{\text{layer}} + d \cdot |V| \cdot c_{\text{matmul}}$ is the full-model per-decode-token cost: L=32 decoder layers at the per-layer total of Table 1 plus the LM head ($|V|$=128,256, ≈3030 s), i.e., ≈53,730 s at n=4,000. For example, `lora_16` saves $12 \cdot (1016.3 + 4.4 + 2.0)/53{,}730 = 22.8\%$. MoD savings do not include the gate, oblivious top-k, or gather/scatter operations, and no figure includes the quality estimator's own cost.

C Extended Discussion

Three observations follow from Tables 2–3 and Figure 5.

Healing flips the dominance order. In Figure 5, every zero-shot configuration (grey crosses) is strictly dominated by the LoRA-healed Pareto frontier. The strongest zero-shot point—scattered `uniK6`, 9.5% MPC saved at 41.6 F1—is dominated by `lora_16 r16`, which saves 2.4× as much MPC (22.8%) at a 10-point higher F1 (51.7). The recovery cost is a one-time training run (rank-16 LoRA on neighbor projections, <13M trainable parameters, ∼28 GPU-minutes), whereas the estimated saving (∼1020 s of MPC time per skipped FFN sub-block per decode token at n=4,000) recurs for every query.

Static skipping and MoD extend the cost range. Stacking fixed-capacity MoD on top of `lora_16` extends the estimated savings from 22.8% to 34.2% (M2–M4 in Table 3) at -1.0 to -6.0 TREC accuracy. These MoD savings count only the skipped FFN computation; the gate, oblivious top-k, and gather/scatter costs are not included, so the net benefit of MoD under MPC remains to be measured. The standalone MoD rows in Table 2 are evaluated on QA rather than TREC and are therefore not directly comparable with M2–M4.

Variant diversity creates routing headroom. On TREC, the oracle realises 79.5% accuracy at an estimated cost of 0.742—7 points above the best single static configuration in the pool. This headroom is a property of the pool: pruned variants disagree on which queries each can answer. The learned estimator does not yet convert it into accuracy: its operating points lie on or below the frontier of fixed variants (Section 4.3), which motivates richer supervision (see Future work below).

Future work. The depth-reduction design transfers in principle to other dense backbones (Mistral, Qwen) and to MoE architectures, where it would compose with CryptoMoE (Zhou et al., 2025) on the orthogonal expert axis. The router would benefit from larger and more diverse labelled corpora, including active labelling on minority query classes and cross-pool training that exposes it to multiple healed pools. End-to-end wall-clock validation on a deployed MPC service, including the router's own secure execution, would tighten the savings estimates obtained from the per-element SPU profiles (Table 4) and the analytical cost model (Appendix B). Finally, deeper attention-skip schedules with attention-side LoRA healing, and MPCache-style token-axis compression *inside* each pruned variant, may further extend the F1-vs-MPC-cost frontier.