HUDA-MoE: Hierarchical Unified Domain-Aligned Mixture of Experts

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) architectures scale large language models efficiently, but standard token-level routers rarely form meaningful domain-specialized computation paths. Existing domain-aware MoE methods typically require oracle domain labels, joint retraining, or careful multi-domain supervision, making post-deployment expansion difficult. We propose **HUDA-MoE**, a frozen-backbone framework for label-free domain specialization with modular LoRA expert banks. HUDA-MoE distills deep-layer K-Means++ clusters into a lightweight Layer-0 domain probe, which selects domain banks at the sequence level before applying token-level routing within the selected banks. This sequence-to-token routing hierarchy separates semantic domain selection from local token allocation while preserving the frozen backbone. HUDA-MoE supports both newly trained experts and external PEFT adapters. Across LLaMA-3.1-8B, Mistral-7B, and Qwen-2.5-7B backbones, **HUDA-MoE** achieves the strongest average downstream performance among controlled baselines under realistic probe-routed evaluation. It further enables near-zero interference expansion, adding a new domain without modifying existing experts and minimal measurable degradation in the original domains.

1 Introduction

Mixture-of-Experts (MoE) architectures have emerged as the dominant paradigm for scaling large language models (LLMs) beyond dense-model limits (Shazeer et al., 2017; Fedus et al., 2022; Jiang et al., 2024). By activating only a sparse subset of expert sub-networks per token, MoE models achieve a compelling efficiency tradeoff. For example, Mixtral-8x7B activates approximately 13B parameters per token from a total of 47B, yet matches or surpasses dense models far above its inference budget (Jiang et al., 2024). At the frontier, DeepSeek-V3 (DeepSeek-AI et al., 2025) scales to 671B total parameters with only 37B active, achieving performance comparable to leading proprietary models.

Despite this impressive scaling, a fundamental question remains unresolved: *Do experts in an MoE system actually specialize meaningfully by knowledge domain?* Empirical investigations consistently find that standard token-level routers do not exhibit clear domain-level specialization (Jiang et al., 2024; Dai et al., 2024). Expert specialization has been identified as essential to take advantage of the increased capacity of sparse MoE models and to prevent redundant knowledge representations (Dai et al., 2024), yet achieving it in practice remains an open challenge (Li et al., 2026). Existing approaches to domain-specialized MoE leave three critical gaps unaddressed: (i) they require *oracle domain labels* at training and inference time (Gururangan et al., 2022; Sukhbaatar et al., 2024; Ma et al., 2024); (ii) they suffer from a *routing granularity mismatch*, routing at the token level where the domain signal is weakest (Yang et al., 2026a; Wen et al., 2026); and (iii) they distribute expert capacity *uniformly across all layers*, ignoring that domain semantics emerge predominantly in deeper transformer layers (Yang et al., 2026b). We discuss these related works and the gaps in more detail in Appendix A.

In this paper, we propose **HUDA-MoE**, a domain-specialization framework that addresses all three gaps simultaneously through *label-free sequence-level domain discovery* and a two-level routing hierarchy over frozen LoRA expert banks. The routing framework requires no domain labels for probe training or inference, although domain-specific LoRA heads are trained on domain-partitioned corpora.

HUDA-MoE is built around three contributions:

(1) Label-Free Hierarchical Routing. A lightweight two-layer MLP (domain probe) is

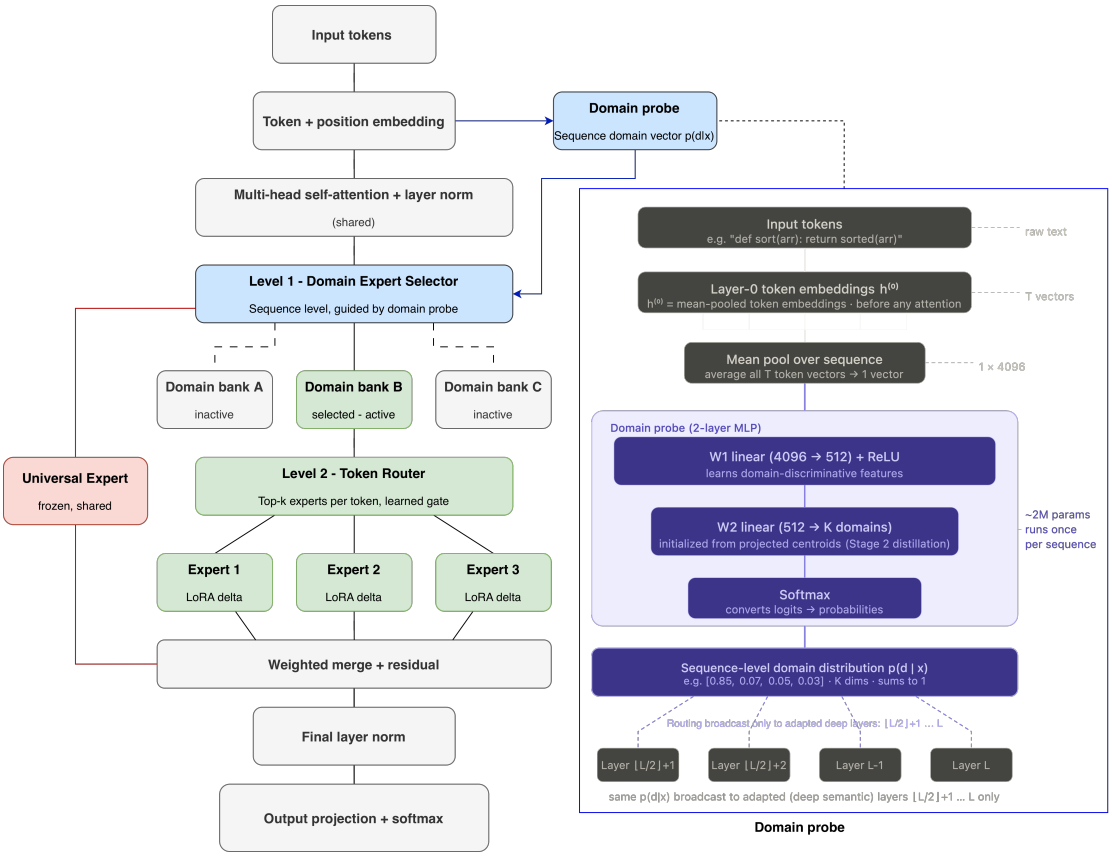

Figure 1: Overview of proposed HUDA-MoE architecture. A lightweight domain probe predicts sequence-level domain distributions from mean-pooled Layer-0 embeddings. These predictions activate one or more domain-specific LoRA expert banks via Level-1 routing, while a Level-2 token router selects experts within each active bank. All experts attach only to a frozen LLM backbone (Universal Expert).

trained on Layer-0 embeddings via cross-layer distillation from deep K-Means++ clusters, discovering latent domain structure without annotation. The probe drives a *Level-1* sequence-level bank selector that activates one or more domain LoRA expert banks per input; within each active bank, a *Level-2* token-level gate selects among experts for each token. This hierarchy decouples domain routing from within-domain token routing, eliminating the granularity mismatch of flat token-level routers.

(2) Asymmetric Depth Schedule. Domain LoRA experts are allocated only at layers $\lfloor L/2 \rfloor + 1$ through L, concentrating capacity in the deep semantic layers where domain-specific representations emerge. A parameter-matched ablation confirms this schedule captures 83.6% of the total perplexity gain of full-depth adaptation while leaving shallow syntactic layers unmodified.

(3) Near-Zero Interference Domain Expansion. Because the backbone is frozen and LoRA banks are independent, adding a new domain post-deployment requires only adding a centroid, updating the lightweight probe, and inserting a new expert bank, with no modifications to any existing experts.

We instantiate HUDA-MoE in two variants: **HUDA-MoE-Lite** (one expert per domain bank, Level-1 soft blending only, lower latency) and **HUDA-MoE** (two experts per bank with a Level-2 token router, stronger performance). We address two research questions: **RQ1:** Does HUDA-MoE outperform flat MoE baselines on downstream tasks under realistic probe-routed conditions across multiple backbone LLMs? **RQ2:** Does the frozen-backbone modular design enable post-deployment domain expansion without modifying existing experts?

Experiments across four domains (Code, Biomed, Math, News) and three backbone LLMs, LLaMA-3.1-8B (Grattafiori et al., 2024), Mistral-7B, and Qwen-2.5-7B, show that **HUDA-MoE** achieves the strongest downstream performance among all controlled baselines under probe-routed inference, and that adding Finance as a fifth domain

post-deployment produces no measurable degradation within our reported evaluation noise. HUDA-MoE-Lite offers a low-latency alternative with only $+6.3\%$ inference overhead over the frozen baseline. We report latency details in Appendix B.

2 Methodology

Figure 1 illustrates the overall HUDA-MoE framework, including the label-free domain probe, hierarchical routing pipeline, frozen universal expert, and domain-specific LoRA expert banks. Our approach adds domain specialization through four synergistic mechanisms: **(1)** an Asymmetric Depth Schedule that concentrates domain capacity in deeper layers, **(2)** a Label-Free Domain Probe that discovers latent domain structure without labels, **(3)** Two-Level Hierarchical Routing that decouples sequence-level domain selection from token-level expert selection, and **(4)** a Joint Fine-Tuning Objective that prevents routing collapse.

2.1 Universal Expert and Asymmetric Depth Schedule

Let L denote the total number of transformer layers in the backbone. All backbone parameters are frozen throughout training and inference. Domain specialization is achieved via Low-Rank Adaptation (LoRA) (Hu et al., 2022) expert modules attached to the attention projection matrices at selected layers.

We partition layers into a *shallow syntactic zone* (layers 1 through $\lfloor L/2 \rfloor$) and a *deep semantic zone* (layers $\lfloor L/2 \rfloor + 1$ through L). Empirical evidence from Yang et al. (2026b) indicates that domain-specific semantic abstraction occurs predominantly in deeper layers, motivating our **Asymmetric Depth Schedule**. In contrast to uniform LoRA placement over all layers, HUDA-MoE assigns no domain experts to shallow layers and allocates all domain-specific expert capacity to the deep semantic zone. Formally, the number of domain LoRA experts per bank at layer l is:

$$N_l = \begin{cases} 0, & l \leq \lfloor L/2 \rfloor, \\ N, & l > \lfloor L/2 \rfloor, \end{cases} \tag{1}$$

Here, N is the number of experts per domain bank in the adapted deep layers: $N = 1$ for HUDA-MoE-Lite and $N = 2$ for HUDA-MoE. This asymmetric deep-half schedule isolates the impact of deep-layer domain adaptation and makes it easy to match the trainable parameter budget to flat MoE baselines. We provide more details on how we set up this design in Appendix C.1.

2.2 Label-Free Domain Probe via Cross-Layer Distillation

Our domain probe is designed for domain-specific text, such as research papers, code files, and clinical documents. To achieve this, we design two stages of pre-training for the probe.

Stage 1 — Deep clustering. Given a corpus of N sequences, we extract *context vectors* from the frozen backbone's final layer:

$$\mathbf{h}_i^{(L)} = \frac{1}{n} \sum_{t=1}^{n} \mathbf{h}_{i,t}^{(L)} \quad \in \mathbb{R}^d \tag{2}$$

where n is the sequence length and $\mathbf{h}_{i,t}^{(L)}$ is the token hidden state at the final layer. In our approach, we use K-Means++ to cluster the vectors. We justify our choice in Appendix C.2.

Stage 2 — Shallow distillation. To enable low-latency routing, we train a lightweight **Domain Probe** on *Layer 0* embeddings (token embeddings before any attention) to predict the Layer-L pseudo-labels:

$$p(d \mid \mathbf{x}) = \mathrm{softmax}\Big(\mathbf{W}_2 \cdot \mathrm{ReLU}\Big(\mathbf{W}_1 \cdot \mathbf{h}^{(0)}\Big)\Big) \tag{3}$$

where $\mathbf{h}^{(0)} = \frac{1}{n}\sum_t \mathbf{h}_t^{(0)} \in \mathbb{R}^d$ is the mean-pooled Layer 0 embedding, $\mathbf{W}_1 \in \mathbb{R}^{d_h \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{K \times d_h}$, and $d_h = 512$. We provide more information on how we train the probe in Appendix C.3.

2.3 Two-Level Hierarchical Routing

HUDA-MoE resolves the *routing granularity mismatch* by operating at two complementary levels: **Level 1** (sequence-level) selects a domain expert bank using the probe, and **Level 2** (token-level) selects among experts within the active bank.

Level-1 — Domain bank selection. The probe output $p(d \mid \mathbf{x})$ is computed once per input sequence from Layer 0 embeddings and broadcast to all adapted layers. The active bank set is:

$$\mathcal{B}(\mathbf{x}) = \{k : p(d{=}k \mid \mathbf{x}) > \tau\} \tag{4}$$

with threshold $\tau = 0.05$ (temperature $T = 10$ applied to $p(d \mid \mathbf{x})$ before thresholding). For unambiguous sequences (typical case), $|\mathcal{B}| = 1$; for domain-ambiguous sequences (e.g., mathematical algorithms), $|\mathcal{B}| = 2$ with proportional blending.

Level-2 — Token-level expert selection. Within the active bank k^*, a learned gate $\mathbf{W}_{\text{gate}}^{(l,k^*)} \in \mathbb{R}^{N_l \times d}$ routes each token's hidden state to one of the N_l experts:

$$\mathbf{g}_t^{(l)} = \text{softmax}\Big(\mathbf{W}_{\text{gate}}^{(l,k^*)} \cdot \mathbf{h}_t^{(l)}\Big) \quad \in \mathbb{R}^{N_l} \tag{5}$$

The gate is applied *per token per layer*, not per sequence. This is the key architectural difference from HUDA-MoE-Lite, which omits Eq. (5) and uses only Level-1 blending. We provide threshold τ and temperature T analyzes in Appendix C.3 and details of the fused forward pass for HUDA-MoE in Appendix C.4.

2.4 Training Objective

HUDA-MoE is trained in three stages: **Stage 1:** domain discovery and routing (Section 2.2), **Stage 2:** LoRA head training (Appendix C.5), and **Stage 3:** fine-tunes the routing components jointly with all LoRA weights frozen, using:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{ent}}\,\mathcal{L}_{\text{ent}} + \lambda_{\text{bal}}\big(\mathcal{L}_{\text{bal}}^{(L1)} + \mathcal{L}_{\text{bal}}^{(L2)}\big) \tag{6}$$

$\mathcal{L}_{\text{LM}}$ is the standard causal language modeling loss. The **entropy regularizer** encourages sharp Level-1 domain assignments:

$$\mathcal{L}_{\text{ent}} = \frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} -p_k^{(i)} \log p_k^{(i)} \tag{7}$$

By minimizing $\mathcal{L}_{\text{ent}}$, we reduce the entropy of $p(d \mid \mathbf{x})$, preventing the probe from hedging across domains.

$\mathcal{L}_{\text{bal}}^{(L1)}$ and $\mathcal{L}_{\text{bal}}^{(L2)}$ are Switch Transformer-style load-balancing losses (Fedus et al., 2022) applied independently at Level-1 (domain bank utilization) and Level-2 (within-bank expert utilization).

2.5 Domain Expansion

In the case of deployment, adding a new domain k^* requires three steps with no modifications to existing expert banks. **(1) Centroid computation:** we extract Layer-L embeddings for new-domain sequences and compute the new centroid $\boldsymbol{\mu}_{k^*}$. Then we can append $\boldsymbol{\mu}_{k^*}$ to the existing centroid matrix, expanding $K \to K+1$. **(2) Probe update:** we retrain the domain probe on a mixed dataset (new-domain sequences plus a small replay buffer from each original domain) with $\mathbf{W}_2$ expanded by one row initialized from the projected centroid. **(3) Expert Insertion:** we either train a new LoRA head or use an existing external LoRA head, keeping all existing heads unchanged. We provide Algorithm 1 and Algorithm 2 in the Appendix C.6 as complete pseudocode for the forward pass of both versions of HUDA-MoE. We also summarize the key differences between versions in Appendix C.7.

3 Experimental Setup

We design experiments to address the two research questions stated in Section 1. All downstream benchmark evaluation uses the `lm-evaluation-harness` (Gao et al., 2024) framework in bfloat16 precision with *probe-routed inference*. The domain probe determines routing at inference time without oracle domain labels to match realistic deployment.

3.1 Backbone LLMs

HUDA-MoE is evaluated on three frozen backbone LLMs to validate architecture-agnosticism: **LLaMA-3.1-8B** (Grattafiori et al., 2024) (L=32, d=4,096), **Mistral-7B** (Jiang et al., 2023) (L=32, d=4,096), and **Qwen-2.5-7B** (Yang et al., 2024) (L=28, d=3,584). LLaMA-3.1-8B is our primary backbone, matching prior work such as MoDULA (Ma et al., 2024) and Med-MoE-LoRA (Yang et al., 2026b); Mistral and Qwen serve as held-out architectures to confirm transferability. All three fit on a single 80 GB A100 with LoRA. The adapted layer range is adjusted per backbone: layers 17–32 for LLaMA and Mistral, layers 15-28 for Qwen, preserving the deep-half schedule of Eq. (1).

3.2 Baseline Configurations (RQ1)

We compare HUDA-MoE against five controlled baselines that share the same backbone, training corpus, adapted layer range, LoRA rank, optimizer, and batch size, grouped by the trainable parameter budget. At ≈27M parameters: **Frozen backbone** (no adaptation), **Flat MoE-4** (four jointly trained LoRA heads with a flat token router), and **LoRA heads + flat router** (independently trained heads, flat token router).

At ≈54M parameters: **Flat MoE-8** (parameter-matched flat baseline for HUDA-MoE) and **Oracle routing** (true domain labels used at inference). Flat MoE routers use the same Switch-style load-balancing objective, and all routing hyper-parameters are selected based on the validation split. Full configuration details are in Appendix D. Three externally pretrained MoE

systems are included as *reference-only* (uncontrolled) points: **OLMoE-1B-7B** (Muennighoff et al., 2025), **DeepSeekMoE-16B** (Dai et al., 2024), and **Mixtral-8x7B** (Jiang et al., 2024). These differ in total parameters, pretraining data, and adaptation procedures.

3.3 Evaluation Benchmarks (RQ1)

We evaluate HUDA-MoE on five downstream benchmarks covering the main domains used in our specialization setup. For code generation, we use **MBPP** (Austin et al., 2021) with 3-shot prompting and report pass@1 on 500 test problems. For biomedical reasoning, we use **MedQA-USMLE** (Jin et al., 2021) in the 0-shot setting (n=1,273) and **PubMedQA** (Jin et al., 2019) in the 0-shot setting (n=500). For mathematical reasoning, we use **GSM8K** (Cobbe et al., 2021) with 5-shot prompting (n=1,319). Finally, we include **MMLU** (Hendrycks et al., 2021a) with 5-shot prompting (n=14,042) as a broad general-knowledge and multi-domain evaluation benchmark.

All results are reported under **probe-routed inference**. The domain probe automatically selects the routing path from the input sequence, without access to benchmark identity or ground-truth domain labels. This prevents benchmark-specific head selection and matches the intended deployment setting, where domain labels are unavailable at inference. The same routing mechanism is used for all controlled baselines to ensure a fair comparison between flat routing, domain-level routing, and the full hierarchical router. For reference, we also report a **domain-matched** setting. This uses oracle head selection based on the known benchmark domain and serves as an upper-bound reference, not the primary evaluation condition. The gap between probe-routed and domain-matched performance shows how much of the oracle domain-selection benefit is recovered by the learned probe. Further details on benchmark construction, prompting, answer extraction, and evaluation metrics are in Appendix E.

3.4 Domain Expansion Protocol (RQ2)

To evaluate RQ2, we expand a HUDA-MoE-Lite from K=4 to K=5 under two independent settings. In both cases, the frozen backbone and all original expert banks remain unchanged; only the centroid matrix and lightweight probe are expanded.

For Finance, we first compute a centroid from Layer-32 embeddings over four financial datasets (Table 14). Then we retrain the probe on 85K mixed sequences with replay and train one Finance LoRA head. For Law, we also compute a centroid using legal text, retrain only the probe with the same replay protocol, and insert the external LegalBench-tuned LoRA as the fifth expert bank. Implementation details are in Appendix F.

4 Results

4.1 Probe-Routed Downstream Benchmarks (RQ1)

Table 1 reports the main results under probe-routed inference across three backbones and controlled baselines. Domain-matched rows are included as an oracle head-selection reference. Under the controlled setting, HUDA-MoE achieves the strongest overall performance for all three backbone LLMs. On LLaMA-3.1-8B, it obtains 53.8 on MBPP, 66.8 on MedQA, 57.2 on GSM8K, 69.5 on MMLU, and 80.7 on PubMedQA, yielding an average score of 65.6. This improves over the frozen backbone by 7.2 points, over Flat MoE-4 by 6.1 points, and over Flat MoE-8 by 3.1 points on average. Flat MoE-4 improves modestly over the frozen backbone, indicating that the flat baseline is functional, but its gains remain limited compared with domain-aware routing.

HUDA-MoE-Lite also performs strongly, reaching 62.2 on average for LLaMA-3.1-8B. It improves over the frozen backbone by 3.8 points, over Flat MoE-4 by 2.7 points, and over LoRA heads with a flat router by 2.8 points, while remaining close to Flat MoE-8. This suggests that sequence-level domain routing already captures much of the benefit of domain specialization, even with a single expert per active bank. The full HUDA-MoE adds token-level expert routing, producing a further 3.4-point gain over HUDA-MoE-Lite on LLaMA-3.1-8B.

Across Mistral-7B and Qwen-2.5-7B, HUDA-MoE averages 63.9 and 65.5, beating Flat MoE-8 by 2.7 and 3.2 points, respectively. Flat MoE-4 achieves small improvements over the frozen model, HUDA-MoE-Lite is competitive with Flat MoE-8, and full HUDA-MoE gives the best controlled result. HUDA-MoE-Lite slightly underperforms Flat MoE-8 on LLaMA-3.1-8B and Qwen-2.5-7B, but slightly outperforms it on Mistral-7B. The domain-matched LLaMA reference scores 66.4, only 0.8 points above controlled HUDA-MoE, showing that the learned router recovers most or-

Table 1: Downstream benchmark results under **probe-routed inference** (our HUDA-MoE). Primary condition for all controlled baselines. **Bold**: best probe-routed result per backbone among controlled baselines. Average column reports mean ± std over 5 independent training seeds for controlled baselines. Δ is the gain over the parameter-matched Flat MoE-8 baseline within the same backbone family. All scores are reported in percentage points (%).

Backbone	Model	MBPP	MedQA	GSM8K	MMLU	PubMedQA	Ave.	Δ
Controlled baselines								
LLaMA-3.1-8B	Frozen backbone	42.8 ± 0.2	60.8 ± 0.3	50.6 ± 0.5	64.4 ± 0.2	73.4 ± 0.4	58.4 ± 0.2	–4.1
	Flat MoE-4 (jointly trained)	46.5 ± 0.8	61.2 ± 0.4	51.0 ± 0.6	65.0 ± 0.5	74.0 ± 0.7	59.5 ± 0.5	–3.0
	LoRA heads + flat router	46.7 ± 0.5	60.9 ± 0.6	50.8 ± 0.9	64.8 ± 0.3	73.7 ± 0.4	59.4 ± 0.4	–3.1
	Flat MoE-8	50.4 ± 0.6	64.1 ± 0.5	53.4 ± 0.8	67.4 ± 0.3	77.2 ± 0.7	62.5 ± 0.4	0.0
	HUDA-MoE-Lite (ours)	50.1 ± 0.4	63.8 ± 0.5	53.0 ± 0.7	67.0 ± 0.4	76.9 ± 0.3	62.2 ± 0.3	–0.3
	HUDA-MoE (ours)	**53.8 ± 0.9**	**66.8 ± 0.4**	**57.2 ± 1.1**	**69.5 ± 0.6**	**80.7 ± 0.5**	**65.6 ± 0.6**	**+3.1**
Mistral-7B	Frozen backbone	41.6 ± 0.5	59.9 ± 0.2	48.5 ± 0.4	63.6 ± 0.3	72.6 ± 0.6	57.2 ± 0.3	–4.0
	Flat MoE-4 (jointly trained)	44.5 ± 0.4	60.9 ± 0.8	48.7 ± 0.7	64.8 ± 0.5	73.9 ± 0.3	58.6 ± 0.5	–2.6
	LoRA heads + flat router	45.2 ± 1.0	60.0 ± 0.3	49.4 ± 0.5	64.0 ± 0.6	72.8 ± 0.8	58.3 ± 0.5	–2.9
	Flat MoE-8	48.7 ± 0.5	62.6 ± 0.7	52.1 ± 0.8	66.5 ± 0.4	76.2 ± 0.6	61.2 ± 0.5	0.0
	HUDA-MoE-Lite (ours)	48.2 ± 0.8	63.2 ± 0.4	52.4 ± 0.6	66.9 ± 0.7	76.1 ± 0.5	61.4 ± 0.5	+0.2
	HUDA-MoE (ours)	**51.0 ± 0.9**	**65.3 ± 0.5**	**55.7 ± 1.0**	**68.5 ± 0.6**	**79.1 ± 0.7**	**63.9 ± 0.6**	**+2.7**
Qwen-2.5-7B	Frozen backbone	41.9 ± 0.2	60.4 ± 0.5	50.4 ± 0.7	65.1 ± 0.3	72.8 ± 0.4	58.1 ± 0.3	–4.2
	Flat MoE-4 (jointly trained)	45.8 ± 0.8	60.8 ± 0.4	51.3 ± 0.9	65.5 ± 0.3	74.1 ± 0.7	59.5 ± 0.5	–2.8
	LoRA heads + flat router	45.0 ± 0.5	61.2 ± 0.6	50.1 ± 0.8	66.1 ± 0.5	73.0 ± 0.4	59.1 ± 0.4	–3.2
	Flat MoE-8	49.2 ± 0.7	63.2 ± 0.4	52.7 ± 1.1	68.9 ± 0.6	77.5 ± 0.5	62.3 ± 0.6	0.0
	HUDA-MoE-Lite (ours)	48.4 ± 0.5	63.7 ± 0.8	53.1 ± 0.7	68.1 ± 0.4	76.6 ± 0.6	62.0 ± 0.5	–0.3
	HUDA-MoE (ours)	**52.6 ± 0.8**	**66.1 ± 0.7**	**56.8 ± 0.9**	**70.8 ± 0.5**	**81.0 ± 0.6**	**65.5 ± 0.6**	**+3.2**
Domain-matched reference (oracle head selection)								
LLaMA-3.1-8B	HUDA-MoE-Lite	50.9 ± 0.5	64.6 ± 0.4	53.8 ± 0.6	67.7 ± 0.3	77.9 ± 0.4	63.0 ± 0.3	
	HUDA-MoE	54.6 ± 0.8	67.6 ± 0.5	58.0 ± 0.9	70.3 ± 0.5	81.6 ± 0.7	66.4 ± 0.5	
Reference-only (uncontrolled, no domain adaptation)								
—	OLMoE-1B-7B	31.8	42.0	48.4	55.1	62.9	48.0	
	DeepSeekMoE-16B	44.5	41.0	49.6	50.4	65.3	50.2	
	Mixtral-8x7B	41.2	61.6	66.4	66.7	77.2	62.6	

acle routing gains. Among reference-only MoE models, controlled HUDA-MoE attains the highest average score, suggesting that improvements come from domain-aware routing and adapter specialization rather than a larger uncontrolled MoE parameter budget. We further analyze our probe and router quality in Appendix G.

4.2 Near-Zero Interference Domain Expansion (RQ2)

We report routing, perplexity, and benchmark results before and after adding a fifth domain to HUDA-MoE-Lite without changing existing expert banks in Table 2. We evaluate two expansion configurations. In the Finance setting, we train a new LoRA expert. In Law, we plug in an externally sourced LegalBench-tuned LoRA adapter. The Finance-expanded probe reaches 99.78% accuracy, which is +2.98 pp higher than the K=4 probe Meanwhile, the Law-expanded probe reaches 99.86%, a +3.06 pp gain. This suggests that the added Finance or Law centroids are well separated from the original domain centroids in the Layer-32 embedding space.

Perplexity on the original four domains remains stable after expansion. Under Finance expansion, Code and Biomed change minimally (−0.01% and −0.13%), while Math and News improve slightly (−0.96% and −0.61%). Under Law expansion, Code and News improve (−0.84% and −0.71%), while Biomed and Math rise slightly (+0.60% and +0.55%). These small, mixed shifts suggest measurement noise rather than systematic interference. In contrast, new-domain perplexity improves substantially. Finance PPL drops by 36.7%, and Law PPL drops by 24.8%, indicating strong domain-specific adaptation without joint fine-tuning.

Downstream performance on original domains remains largely unchanged. Finance expansion keeps all shifts within ±0.5 pp, with small gains on MBPP, GSM8K, and PubMedQA, and no change on MedQA or MMLU. Law expansion similarly

Table 2: Near-zero interference domain expansion in HUDA-MoE-Lite. We evaluate two independent expansion settings from the same deployed HUDA-MoE-Lite checkpoint. Finance evaluates the standard setting in which a new LoRA expert is trained by us. Law evaluates external-adapter expansion. In both settings, no existing banks are modified. PPL is evaluated through the updated probe. Negative Δ in PPL indicates improvement; positive Δ in benchmarks indicates improvement.

Metric	**Pre-expansion**	**Finance expansion: trained LoRA**		**Law expansion: external LoRA**	
		Post-expansion	Δ	**Post-expansion**	Δ
Routing					
Average routing accuracy	96.8%	99.78%	+2.98 pp	99.86%	+3.06 pp
Perplexity (original four domains)					
Code PPL	2.496	2.495	−0.01%	2.475	−0.84%
Biomed PPL	5.643	5.635	−0.13%	5.677	+0.60%
Math PPL	5.484	5.431	−0.96%	5.514	+0.55%
News PPL	7.176	7.132	−0.61%	7.125	−0.71%
New domain PPL(Finance/Law)	9.024/15.263	5.710	−36.7%	11.479	−24.8%
Downstream benchmarks (original domains)					
MBPP	50.1	50.5	+0.4 pp	49.9	−0.2 pp
MedQA	63.8	63.8	+0.0 pp	63.9	+0.1 pp
GSM8K	53.0	53.5	+0.5 pp	52.8	−0.2 pp
MMLU	67.0	67.0	+0.0 pp	67.1	+0.1 pp
PubMedQA	76.9	77.0	+0.1 pp	77.1	+0.2 pp

causes insignificant changes across the five benchmarks. Thus, HUDA-MoE-Lite supports post-deployment extensibility. **New domains can be added via independently trained LoRA experts or external adapters while preserving existing experts and original-domain behavior.**

5 Discussion

5.1 Routing Architecture Ablation

In Table 3, we report per-domain validation perplexities for all six LLaMA-3.1-8B configurations. HUDA-MoE-Lite attains an average PPL of 5.200, reflecting a **6.5%** decrease relative to the frozen baseline under matched parameter counts. Flat MoE-4 and LoRA heads with a flat router also improve over the frozen baseline, reaching average PPLs of 5.464 and 5.454, respectively. However, their gains are modest, and their routing accuracies remain much lower than HUDA-MoE-Lite (63.7% and 65.1% versus 96.8%), suggesting that flat routing captures only partial domain specialization. This indicates that the performance gap stems from the routing architecture rather than from the availability of expert heads alone. The full HUDA-MoE model reaches an average PPL of 4.900, corresponding to an **11.9%** reduction versus the frozen baseline and a **4.2%** improvement over the parameter-matched Flat MoE-8. Its strongest per-domain improvements appear in the News and Biomed domains.

5.2 Asymmetric Depth Schedule

We compare deep-only and uniform LoRA under matched trainable parameters in Table 4. Deep-only adaptation (layers 17-32) accounts for **83.6%** of the total perplexity gain over the frozen baseline, with only 16.4% coming from layers 1–16. This aligns with prior work on transformer layer specialization (Tenney et al., 2019; Geva et al., 2021): shallow layers capture syntactic and surface features shared across domains, while deeper layers encode domain-specific semantics. Concentrating LoRA rank in the latter is therefore a principled design choice, not a parameter-saving heuristic.

The schedule generalizes across architectures: applying the deep-half rule to Mistral-7B and Qwen-2.5-7B with identical hyper-parameters yields consistent in-domain PPL reductions and near-zero OOD degradation on Code, Math, and News. The Biomed head shows 13–22% OOD degradation across all three backbones since biomedical vocabulary pushes deep layers further from the general pretraining distribution. Full multi-architecture PPL results are in Table 17 (Appendix H).

5.3 What Does the Level-2 Router Learn?

Table 5 shows that all four domain banks lie in the healthy entropy range ($0.10 < H < 0.60$). This indicates that the Level-2 router makes confident, non-trivial per-token decisions rather than collapsing to uniform or single-expert routing. The routing

Table 3: Per-domain validation perplexity across all six configurations (LLaMA-3.1-8B). This experiment uses the validation sets in same corpus when training LoRA heads (Table 14). Lower is better, best result per column in **bold**.

	Perplexity ↓					
Model	**Code**	**Biomed**	**Math**	**News**	**Avg**	**Rout. Acc.**
Frozen LLaMA-3.1-8B	2.599	6.236	5.656	7.763	5.563	—
Flat MoE-4 (jointly trained)	2.548	6.104	5.603	7.601	5.464	63.7%
LoRA heads + flat router	2.561	6.081	5.624	7.548	5.454	65.1%
HUDA-MoE-Lite (ours)	**2.496**	**5.643**	**5.484**	**7.176**	**5.200**	96.8%
Flat MoE-8	2.460	5.589	5.319	7.100	5.117	71.4%
HUDA-MoE (ours)	**2.410**	**5.381**	**5.081**	**6.727**	**4.900**	99.96%
Oracle routing	2.410	5.380	5.078	6.720	4.897	100%

Table 4: Parameter-matched depth ablation on the Code domain. Both configurations have exactly 6,815,744 trainable parameters. Lower PPL is better.

Configuration	**Layers**	**Rank**	**Params**	**PPL ↓**
Frozen baseline	—	—	0	2.627
Deep-only (ours)	17–32	16	6,815,744	2.489
Uniform LoRA	1–32	8	6,815,744	**2.462**
Benefit captured by deep-only				83.6%

distributions are asymmetric. In the Math bank, Expert 1 dominates (79.0%), as the pre-trained head already captures most mathematical notation, while Expert 2 handles the residual prose. In the Code bank, our router reverses Expert 2 at 74.8%, suggesting that it focuses on comments and documentation. This asymmetry demonstrates expert divergence rather than collapse and accounts for HUDA-MoE's 4.2% PPL gain over the parameter-matched Flat MoE-8.

Table 5: Level-2 within-bank routing entropy for HUDA-MoE. $H = \ln 2 \approx 0.693$ is the uniform ceiling. All domains fall within the healthy range ($0.10 < H < 0.60$).

Domain	**Entropy** H	$H/\ln 2$	**Expert 1 / Expert 2**
Code	0.335	0.483	25.2% / 74.8%
Biomed	0.389	0.561	23.6% / 76.4%
Math	0.322	0.466	79.0% / 21.0%
News	0.424	0.612	64.1% / 35.9%
Collapsed	0.693	1.000	50% / 50%

5.4 Domain-Ambiguous Routing

In Table 6, we report the per-domain routing accuracy and the average number of active banks for HUDA-MoE-Lite. The router achieves high accuracy for Code, Biomed, and News, with scores between 98.3% and 98.9%, while Math has a lower routing accuracy of 91.2%.

Table 6: Per-domain routing accuracy and average active banks for HUDA-MoE-Lite (τ=0.05, T=10). Math activates nearly two banks on average, reflecting genuine domain overlap with Code.

Domain	**Routing Acc.**	**Avg Active Banks**
Code	98.9%	1.20
Biomed	98.9%	1.16
Math	91.2%	1.96
News	98.3%	1.23
Overall	96.8%	1.39

Math activates the most banks on average, with 1.96 active banks versus about 1.2 for other domains. Overall, HUDA-MoE-Lite attains 96.8% routing accuracy with 1.39 active banks per sequence, indicating sparse routing while enabling more multi-bank activation for Math. Domain probe confusion matrix analysis is in Appendix I.

6 Conclusion

We introduced **HUDA-MoE**, a hierarchical, parameter-efficient MoE framework that resolves the routing granularity mismatch of token-routed MoEs via label-free sequence-level domain discovery and two-level hierarchical routing over a frozen LLM backbone. HUDA-MoE reduces domain perplexity, improves downstream benchmarks, and preserves general-domain performance. Under probe-routed inference, HUDA-MoE improves on parameter-matched flat MoE baselines across three backbone LLMs, indicating that the gains stem from the routing architecture rather than added capacity. Finance and Law expansion experiments show that new domains can be added either by training a new LoRA expert or reusing an external PEFT adapter while preserving existing expert banks.

Limitations

Probe scope. The probe achieves 96.8% routing accuracy on long-form domain text and shows reliable routing behavior on most downstream benchmarks. However, it remains a lightweight document-level router rather than a task-specific prompt classifier. Consequently, routing can still be imperfect for short or highly structured prompts whose surface form differs from the long 2,048-token sequences used during probe training. For example, mathematical notation, code-like symbols, and multiple-choice formatting may blur domain boundaries at the Layer-0 representation level. These cases suggest that future work could combine document-level routing with prompt-format-aware calibration or late-layer task signals while preserving the low overhead of the current HUDA-MoE probe.

Expert training still requires domain-partitioned data. While the domain probe is label-free, expert capacity in the standard setting comes from LoRA heads trained on domain-partitioned corpora. HUDA-MoE thus removes the need for domain labels at routing time, but it does not remove the need for a domain partition when training new experts. The only exception is when an externally trained adapter is reused, as in our Law expansion experiment.

Scope of the domain and expansion settings. Our demonstration experiments use only four domains. Routing among more semantically adjacent domains, where surface features overlap, is left for future work. Similarly, near-zero interference expansion is demonstrated for a single $K{=}4 \rightarrow 5$ step under two settings. How the system behaves under numerous expansions remains unresolved.

Constraint on model size. Controlled backbone baselines are evaluated for three 7B/8B-class models only. Larger backbones and instruction-tuned variants remain future work. External MoE systems differ in pretraining data, parameter count, and adaptation procedures, so they are included only as reference points.

Ethics Statement

This work studies the parameter-efficient domain specialization of large language models and involves no human subjects or new data collection. All datasets and pre-trained models are publicly available and used under their respective licenses. We did not re-annotate them, so any biases or inaccuracies in the original sources are inherited by the adapted models. The biomedical and legal text domains are sensitive. HUDA-MoE improves in-domain accuracy but does not verify factual correctness, and our results are intended to demonstrate routing and adaptation behavior, not to support clinical or legal decision-making. Because the backbone is frozen, HUDA-MoE preserves its original behavior on non-adapted inputs and does not weaken any existing alignment. It could be used to specialize a model toward harmful content, but it introduces no misuse risk beyond that of the underlying models. All experiments run on single A100/H100 GPUs and train only LoRA experts and a lightweight probe, substantially reducing compute relative to full fine-tuning. Hardware, hyperparameters, dataset sources, and evaluation protocols are reported in the appendices to support reproducibility.

Acknowledgments

References

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *CoRR*, abs/2108.07732.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

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. 2024. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, Bangkok, Thailand. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, and et al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Peter Henderson, Mark S. Krass*, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *arXiv preprint*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021b. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, and et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, and et al. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Jiaang Li, Haibin Chen, Langming Liu, Yujin Yuan, Yadao Wang, Yizhen Zhang, Chengting Yu, Xin Tong, Weidong Zhang, Shilei Liu, Wenbo Su, and Bo Zheng. 2026. Expert divergence learning for moe-based language models. In *The Fourteenth International Conference on Learning Representations*.

Junzhuo Li, Bo Wang, Xiuze Zhou, and Xuming Hu. 2025. Dynamic expert specialization: Towards catastrophic forgetting-free multi-domain MoE adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18478–18493, Suzhou, China. Association for Computational Linguistics.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, and 7 others. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Minxuan Lv, Zhenpeng Su, Leiyu Pan, Yizhe Xiong, Zijia Lin, Hui Chen, Wei Zhou, Jungong Han, Guiguang Ding, Wenwu Ou, Di Zhang, Kun Gai, and Songlin Hu. 2025. DSMoE: Matrix-partitioned experts with dynamic routing for computation-efficient dense LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19711–19722, Suzhou, China. Association for Computational Linguistics.

Yufei Ma, Zihan Liang, Huangyu Dai, and et al. 2024. MoDULA: Mixture of domain-specific and universal LoRA for multi-task learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2758–2770, Miami, Florida, USA. Association for Computational Linguistics.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and

5 others. 2025. OLMoe: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *Preprint*, arXiv:2310.06786.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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

Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Roziere, Jacob Kahn, Shang-Wen Li, Wen tau Yih, Jason E Weston, and Xian Li. 2024. Branch-train-mix: Mixing expert LLMs into a mixture-of-experts LLM. In *First Conference on Language Modeling*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Tiansheng Wen, Yifei Wang, Aosong Feng, Long Ma, Xinyang Liu, Yifan Wang, Lixuan Guo, Bo Chen, Stefanie Jegelka, and Chenyu You. 2026. Route experts by sequence, not by token. *Preprint*, arXiv:2511.06494.

An Yang, Baosong Yang, and et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Shengtian Yang, Yu Li, Shuo He, Yewen Li, Qingpeng Cai, Peng Jiang, and Lei Feng. 2026a. Phase-aware mixture of experts for agentic reinforcement learning. *Preprint*, arXiv:2602.17038.

Yuxin Yang, Aoxiong Zeng, and Xiangquan Yang. 2026b. Towards specialized generalists: A multi-task moe-lora framework for domain-specific llm adaptation. *Preprint*, arXiv:2601.07935.

A Related Work

A.1 Mixture-of-Experts Architectures and Routing

Sparse MoE architectures scale model capacity by routing each token to a learned subset of expert sub-networks (Shazeer et al., 2017). The Switch Transformer (Fedus et al., 2022) introduced the top-1 routing and load-balancing loss that is now standard across MoE systems. Mixtral-8×7B (Jiang et al., 2024) demonstrated that top-2 routing over eight FFN experts matches dense models far above its 13B active-parameter inference budget, while DeepSeek-V3 (DeepSeek-AI et al., 2025) scales to 671B total parameters with 37B active. Despite this scaling success, standard token-level routers do not exhibit clear domain-level specialization (Jiang et al., 2024), because domain identity is a property of entire sequences rather than individual tokens. PA-MoE (Yang et al., 2026a) and SeqTopK (Wen et al., 2026) independently validate this routing-granularity mismatch in adjacent settings, but neither addresses multi-domain LLM specialization or unsupervised domain discovery.

A.2 Domain-Specialised Mixture of Experts

DEMix (Gururangan et al., 2022) showed that conditioning expert FFN layers on oracle domain labels reduces test-time perplexity and enables rapid domain adaptation. Branch-Train-MiX (Sukhbaatar et al., 2024) scales this by training parallel domain-specialist LLMs and merging their parameters into a unified MoE, but it requires full pre-training per domain, and the post-merge token-level router inherits the granularity mismatch. MoDULA (Ma et al., 2024) takes a parameter-efficient route with universal and domain-specific LoRA adapters, reducing training costs by over 80%, yet it still relies on manual domain partitioning. Lv et al. (2025) provides empirical evidence that domain-specific experts exist latently in trained MoE systems and proposes a training-free steering framework, but it cannot grow the expert pool for new domains.

A.3 LoRA-Based Mixture of Experts

Mixing LoRA adapters as MoE experts has emerged as a parameter-efficient alternative to full-weight MoE (Hu et al., 2022). MoDULA-Res (Ma et al., 2024) connects universal and domain-specific LoRA adapters via a residual path, but routes at the token level and requires labeled domain partitioning. Med-MoE-LoRA (Yang et al., 2026b)

Table 7: Comparison with domain-specialized and sequence-routed MoE methods.

Method	Label-free routing	Seq-level routing	Token-level routing	Expansion
DEMix (Gururangan et al., 2022)	×	✓	×	limited
MoDULA (Ma et al., 2024)	×	×	✓	×
Branch-Train-MiX (Sukhbaatar et al., 2024)	×	×	✓	costly
SeqTopK (Wen et al., 2026)	✓	✓	×	×
Med-MoE-LoRA (Yang et al., 2026b)	×	×	✓	×
DSMoE (Lv et al., 2025)	✓	×	✓	×
DES-MoE (Li et al., 2025)	×	Partial	✓	Partial
HUDA-MoE (ours)	✓	✓	✓	✓

demonstrates that allocating more LoRA experts to deeper transformer layers improves domain task performance in clinical NLP, validating our Asymmetric Depth Schedule in a single-domain context, but does not generalize to multi-domain routing or architecture-agnostic evaluation.

A.4 Positioning against prior work.

DEMix (Gururangan et al., 2022) and Branch-Train-MiX (Sukhbaatar et al., 2024) require oracle domain labels at both training and inference times. MoDULA (Ma et al., 2024) reduces pretraining costs through universal and domain-specific LoRA adapters but retains manual domain partitioning and flat token-level routing. Sequence-level routing works, such as SeqTopK (Wen et al., 2026) and PA-MoE (Yang et al., 2026a), addressing the routing granularity mismatch in adjacent settings, but neither combines sequence-level routing with unsupervised domain discovery nor supports post-deployment domain expansion. Med-MoE-LoRA (Yang et al., 2026b) validates depth-aware LoRA allocation in a single-domain clinical setting without generalizing to multi-domain routing or architecture-agnostic evaluation. To our knowledge, **HUDA-MoE** is the first system to unify **label-free domain discovery**, a **sequence-to-token routing hierarchy**, and **modular post-deployment expansion** in a single frozen-backbone framework. DSMoE (Lv et al., 2025) partitions the FFN matrices of a pretrained dense LLM into computational blocks and learns dynamic sparse routing over them for compute-efficient dense-to-MoE conversion, not domain-specialized adaptation. It does not support label-free domain discovery or modular expert addition. DES-MoE (Li et al., 2025) is closer to our setting. It targets multi-domain MoE adaptation and mitigates catastrophic forgetting via adaptive routing, expert–domain correlation tracking, and progressive parameter freezing. But it assumes domain-partitioned data and only updates experts in an existing MoE. In contrast, HUDA-MoE discovers routing without domain labels for the probe, uses frozen-backbone LoRA expert banks, and adds new domains by training separate banks without modifying existing experts. Table 7 compares our approach with previous work.

B Latency and Memory Overhead Measurement

HUDA-MoE-Lite and full HUDA-MoE occupy distinct points on the latency–performance curve, making the choice of variant a deployment decision rather than a correctness one. Table 8 reports wall-clock measurements on an A100-80GB for 2,048-token sequences. The domain probe itself adds only 3.78 ms (+1.9%) over the frozen baseline, confirming that sequence-level routing adds low overhead in our A100 setup. HUDA-MoE-Lite with a single active bank adds 12.65 ms (+6.3%) with no additional memory, making it suitable for latency-sensitive deployment settings. Full HUDA-MoE adds 22.10 ms (+11.0%) and 2.2 GB of peak GPU memory for the second expert bank, trading this overhead for the stronger downstream performance reported in Table 1. Crucially, multi-bank activation does not double latency. The two-bank HUDA-MoE measured overhead is 23.30 ms, only 1.20 ms above two-bank HUDA-MoE-Lite. Therefore, the practical deployment latency of HUDA-MoE-Lite remains within the single-bank budget for virtually all practical workloads.

C Details on Methodology

C.1 Setup for Universal Expert and Asymmetric Depth Schedule

HUDA-MoE uses a binary depth-asymmetric allocation. No domain LoRA experts are attached to the shallow half of the backbone, while a constant number of experts are attached to each layer in the

Table 8: Forward-pass latency and peak GPU memory per 2,048-token input sequence. Latency is measured for a single teacher-forced forward pass, not auto-regressive decoding. Throughput is computed as input tokens divided by forward-pass latency. Overhead is relative to the frozen baseline. KV-cache is 0.25 GB for all configurations.

Configuration	Latency (ms)	Overhead	Peak GPU (GB)	Tok/s
Frozen LLaMA-3.1-8B	201.73 ± 2.29	—	15.99	10,152
+ Probe only	205.51 ± 1.03	+3.78 ms (+1.9%)	16.00	9,965
HUDA-Lite (1 bank)	214.38 ± 1.52	+12.65 ms (+6.3%)	16.00	9,553
HUDA-Lite (2 banks)	225.03 ± 2.53	+23.30 ms (+11.5%)	18.19	9,101
Full HUDA	223.83 ± 1.61	+22.10 ms (+11.0%)	18.19	9,150

Table 9: Clustering algorithm comparison on Layer 32 embeddings (200K full scale; 10K aligned subsample for Spectral). HDBSCAN naturally discovers K=2 at 200K scale, while Spectral clustering requires a 10K aligned subsample due to scalability constraints.

Algorithm	ARI	AMI	Silhouette	Runtime (s)	Centroids	Scalable
K-Means++	**0.926**	**0.902**	**0.239**	268	✓	✓
K-Medoids	0.919	0.893	0.238	155	✓	✓
Bisecting K-Means	0.901	0.883	0.237	**23**	✓	✓
GMM (diagonal)	0.916	0.902	0.236	430	✓	✓
HDBSCAN (K=2 found)	0.323	0.555	0.246	116	×	✓
Spectral (10K subsample)	0.696	0.814	0.160	25	×	×

deep half. This design isolates the effect of placing domain-specific capacity in deeper semantic layers rather than introducing an additional layer-wise expert-allocation hyperparameter.

In HUDA-MoE-Lite, we set $N = 1$ for every adapted deep layer. Thus, each domain bank contains one LoRA expert at layers $\lfloor L/2 \rfloor + 1$ through L, and no domain expert elsewhere. In full HUDA-MoE, we set $N = 2$ for every adapted deep layer, enabling the Level-2 router (§2.3) to learn within-bank token-level specialization. For LLaMA-3.1-8B ($L = 32$), the adapted range is layers 17–32 in both variants. For Mistral-7B ($L = 32$), we also adapt layers 17–32, while for Qwen-2.5-7B ($L = 28$), we adapt layers 15–28.

Each LoRA expert in domain bank k at layer l is parameterized by matrices $\mathbf{A}_{k,j}^{(l)} \in \mathbb{R}^{r \times d}$ and $\mathbf{B}_{k,j}^{(l)} \in \mathbb{R}^{d_{\text{out}} \times r}$, where j indexes the within-bank expert and $r \ll d$. The LoRA modules are applied to the query, key, value, and output projection matrices of the self-attention block. We set $r = 16$, $\alpha = 32$, and dropout$= 0.05$ throughout.

C.2 Justifying the use of K-Means++

To justify the K-Means++ choice, Table 9 compares six clustering algorithms on the same 200K Layer 32 embeddings. K-Means++ achieves the highest ARI among algorithms that produce direct centroid vectors (0.926), the architectural requirement enabling $\mathbf{W}_2$ initialization and epoch-0 probe convergence. HDBSCAN achieves ARI = 0.950 on a 10K subsample but collapses to ARI = 0.323 on the full 200K corpus—discovering only two clusters with 13% noise points—making it unreliable for production-scale deployment. We conclude that K-Means++ is the Pareto-optimal choice: the highest clustering quality among centroid-producing algorithms, deterministic output, and linear scalability.

Ablation experiments on different K values: Table 10 reports Layer-32 K-Means++ ARI across K $\in \{2, 3, 4, 5, 6, 7, 8\}$ on the 8K corpus. Layer-32 ARI peaks sharply at K=4 and degrades monotonically thereafter (K=5: 0.875; K=6: 0.785; K=8: 0.719), confirming that the backbone deep representations naturally partition into four clusters without manual tuning. These results confirm that K=4 is not an artificially privileged choice but the natural cluster count supported by the backbone internal geometry.

C.3 Probe Training

Stage 1: We apply K-Means++ with K clusters and $n_{\text{init}} = 10$ restarts to obtain K centroids $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ and pseudo-labels $y_i = \arg\min_k \|\mathbf{h}_i^{(L)} - \boldsymbol{\mu}_k\|_2$.

Stage 2: We initialize $\mathbf{W}_2$ by projecting the K-Means centroids $\{\boldsymbol{\mu}_k\}$ through $\mathbf{W}_1$ and normal-

Table 10: K-sweep ARI for Layer-32 and Layer-0 K-Means++ on the 8K POC corpus. Bold: K=4 (used in HUDA-MoE).

Representation	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Layer-32	0.320	0.666	**0.934**	0.875	0.785	0.746	0.719
Layer-0	0.309	0.355	**0.665**	0.722	0.756	0.760	0.732
TF-IDF	0.271	0.602	**0.807**	0.803	0.752	0.694	0.647

Table 11: Threshold sensitivity at fixed temperature $T = 10$.

τ	Routing Acc.	Avg. Active Banks	Avg. PPL ↓
0.01	96.6	1.71	5.207
0.03	96.9	1.52	5.194
0.05	**96.8**	**1.39**	**5.200**
0.07	96.3	1.24	5.228
0.10	95.7	1.12	5.261

Table 12: Temperature sensitivity at fixed threshold $\tau = 0.05$.

T	Routing Acc.	Avg. Active Banks	Avg. PPL ↓
1	95.9	1.08	5.248
5	96.5	1.27	5.216
10	**96.8**	**1.39**	**5.200**
15	96.6	1.55	5.209
20	96.2	1.72	5.237

izing, so the probe achieves near-optimal routing before any gradient update. The probe comprises ≈2.1M parameters in total. It is trained with cross-entropy loss against pseudo-labels $\{y_i\}$, using Adam ($\eta = 10^{-3}$, weight decay 10^{-5}), batch size 256. Layer 0 extraction requires approximately 2 min on an A100-40GB for 200K sequences, versus 290 min for Layer L, a $145\times$ speed-up that makes the probe practical for rapid domain expansion.

Routing threshold and temperature sensitivity.

Tables 11 and 12 show that HUDA-MoE is relatively stable around the default routing hyperparameters. For the threshold analysis, very small τ values activate more LoRA banks and slightly improve recall for ambiguous examples, but they also increase routing density without clear perplexity gains. In contrast, overly large τ values reduce the number of active banks but slightly degrade routing accuracy, suggesting that some ambiguous-domain examples benefit from multi-bank activation. The default $\tau = 0.05$ provides a good trade-off between routing accuracy, sparsity, and perplexity.

A similar trend appears in the temperature experiment. Low temperatures produce sharper routing distributions and activate fewer banks, while high temperatures soften the distribution and increase the number of active banks. The default $T = 10$ achieves the best overall balance, giving the lowest perplexity while maintaining high routing accuracy and moderate routing sparsity.

C.4 Fused Forward Pass

The complete hidden state update at layer l for token t combines the frozen backbone and the active LoRA experts:

HUDA-MoE-Lite (single expert per bank, $N_l = 1$):

$$\mathbf{h}_t^{(l)} = \underbrace{\mathbf{W}_0^{(l)}\mathbf{x}_t}_{\text{frozen backbone}} + \frac{\alpha}{r} \underbrace{\sum_{k \in \mathcal{B}(\mathbf{x})} p_k \cdot \mathbf{B}_k^{(l)} \mathbf{A}_k^{(l)} \mathbf{x}_t}_{\text{soft LoRA mixture}} \tag{8}$$

HUDA-MoE ($N_l \geq 1$ experts per bank):

$$\mathbf{h}_t^{(l)} = \mathbf{W}_0^{(l)}\mathbf{x}_t + \frac{\alpha}{r} \sum_{k \in \mathcal{B}(\mathbf{x})} p_k \underbrace{\sum_{j=1}^{N_l} g_{t,j}^{(l)} \cdot \mathbf{B}_{k,j}^{(l)} \mathbf{A}_{k,j}^{(l)} \mathbf{x}_t}_{\text{Level-2 expert mixture}} \tag{9}$$

where $\mathbf{x}_t = \mathbf{h}_t^{(l-1)}$, $\mathbf{W}_0^{(l)}$ is the frozen backbone weight matrix, $p_k = p(d{=}k \mid \mathbf{x})$ is the Level-1 routing weight, and $g_{t,j}^{(l)}$ is the Level-2 token routing weight from Eq. (5). Both equations apply independently to each of the four target projection matrices ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$). LoRA contributions from banks with $p_k < \tau$ are pruned.

In Eq. (8), Level-1 routing subsumes Level-2: with $N_l = 1$, the inner sum degenerates to a single expert, making HUDA-MoE-Lite a special case of Full HUDA-MoE.

C.5 LoRA Training

LoRA Configuration The same LoRA configuration applies to all domain expert heads:

Training Corpus Table 14 summarizes the corpora used for the original four-domain training and

Table 13: LoRA hyperparameters shared across all domain expert heads.

Hyperparameter	Value
Rank (r)	16
Alpha (α)	32
Dropout	0.05
Target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`
Target layers	17–32 (Asymmetric Depth Schedule, Eq. 1)
Trainable params / head	6,815,744
Optimiser	AdamW (β_1=0.9, β_2=0.999)
Learning rate	2×10^{-4} with cosine decay
Warmup steps	3% of total steps
Weight decay	0.01
Effective batch size	16 (batch = 4, gradient accumulation = 4)
Max epochs	3
Precision	bf16 (LoRA weights)

domain expansion. All documents are tokenized into fixed 2,048-token sequences without padding and shuffled with 5 seeds. The original training corpus covers Code, Biomedicine, Math, and News, drawing from widely used Hugging Face datasets. For expansion experiments, Finance is used to train a newly added domain expert. Law datasets are used for probe training and evaluation.
Each domain uses 45K sequences for training and 5K for validation during LoRA head training. Domain probe training uses 160K sequences for training and 40K for testing, with a further 90/10 split of the training set for probe validation.

C.6 Pseudocode for HUDA-MoE-Lite

Below are the Algorithm 1 and Algorithm 2 that we provide for HUDA-MoE and HUDA-MoE-Lite:

C.7 HUDA-MoE-Lite versus HUDA-MoE: Key Differences

Table 15 summarizes the architectural differences between the two variants.

D Evaluation Configurations

Below are our model configurations. To support controlled comparisons, several configurations reuse the identical four independently-trained LoRA domain head checkpoints (one per domain; r=16, layers 17–32, trained separately on domain-partitioned corpora). We refer to this fixed set as the shared domain heads. The configurations that use it differ *only* in the routing module attached on top. It is what makes the routing hierarchy isolatable.

Frozen LLaMA-3.1-8B. The base model with no LoRA adaptation and no routing. Establishes the ceiling for general capability and the floor for domain-specific adaptation.

Algorithm 1 HUDA-MoE Forward Pass

Require: Token embeddings $\mathbf{X} \in \mathbb{R}^{n \times d}$, frozen backbone $\{\mathbf{W}_0^{(l)}\}$, LoRA expert pairs $\{(\mathbf{A}_{k,1}^{(l)}, \mathbf{B}_{k,1}^{(l)}, \mathbf{A}_{k,2}^{(l)}, \mathbf{B}_{k,2}^{(l)})\}$, Level-2 gates $\{\mathbf{W}_{\text{gate}}^{(l,k)}\}$, domain probe f_ϕ, threshold τ
Ensure: Output hidden states $\mathbf{H}^{(L)}$

```
// Level-1: Sequence-level routing (once per sequence)
```
$\bar{\mathbf{h}}^{(0)} \leftarrow \frac{1}{n}\sum_t \mathbf{X}[t,:]$
$\mathbf{p} \leftarrow f_\phi\left(\bar{\mathbf{h}}^{(0)}\right)$
$\mathcal{B} \leftarrow \{k : p_k > \tau\}$
```
// Layer-wise forward pass
for l = 1 to L do
```
$\mathbf{X} \leftarrow \mathrm{LN}\left(\mathbf{X} + \mathrm{MHSA}^{(l)}(\mathbf{X})\right)$
$\Delta\mathbf{X} \leftarrow \mathbf{0}$
```
    if l > ⌊L/2⌋ then
        for k ∈ B do
            // Level-2: Per-token routing within active bank
```
            for $t = 1$ to n do
$\mathbf{g}_t \leftarrow \mathrm{softmax}\left(\mathbf{W}_{\text{gate}}^{(l,k)}\mathbf{X}[t,:]\right) \in \mathbb{R}^2$
$\delta_t \leftarrow g_{t,1} \cdot \mathbf{X}[t,:]\mathbf{A}_{k,1}^{(l)\top}\mathbf{B}_{k,1}^{(l)\top} + g_{t,2} \cdot \mathbf{X}[t,:]\mathbf{A}_{k,2}^{(l)\top}\mathbf{B}_{k,2}^{(l)\top}$
$\Delta\mathbf{X}[t,:] \mathrel{+}= p_k \cdot \frac{\alpha}{r}\,\delta_t$
```
            end for
        end for
    end if
```
$\mathbf{X} \leftarrow \mathbf{X} + \mathrm{FFN}^{(l)}(\mathbf{X}) + \Delta\mathbf{X}$
```
end for
return X
```

Flat MoE-4 (jointly trained). Four LoRA heads (r=16, layers 17–32) trained *jointly* on the full mixed corpus (25% per domain per batch) with a learned token-level routing gate $\mathbf{W}_{\text{gate}}$ and a Switch Transformer load-balancing loss. These heads are *not* the shared domain heads; they are co-adapted under the joint objective. Represents the flat MoE baseline at the ≈27M-parameter budget.

Table 14: Corpus composition for original training and domain expansion. All examples are tokenized into 2,048-token sequences with no padding and shuffled with 5 seeds. Finance is used for the newly trained expansion expert, while Law is used for probe training and evaluation with an externally obtained Law adapter.

Domain	Dataset	Source	Sequences	Tokens
Code	`codeparrot/codeparrot-clean`	HuggingFace	50,000	102M
Code	`deepmind/code_contest` (Li et al., 2022)	Github	3,204	6M
Biomed	`ccdv/pubmed-summarization` (Cohan et al., 2018)	HuggingFace	50,000	102M
Biomed	`openlifescienceai/medmcqa` (Pal et al., 2022)	HuggingFace	6,285	12.8M
Math	`open-web-math/open-web-math` (Paster et al., 2023)	HuggingFace	50,000	102M
News	`cc_news+ cnn_dailymail` (See et al., 2017)	HuggingFace	50,000	102M
Total original 4 domains			**209,489**	**∼429M**
Finance†	`zeroshot/twitter-financial-news-topic`	HuggingFace	45,000	92M
	`nickmuchi/financial-classification`	HuggingFace		
	`gbharti/finance-alpaca`	HuggingFace		
	`sujet-ai/Sujet-Finance-Instruct-177k`	HuggingFace		
Law‡	`theatticusproject/cuad` (Hendrycks et al., 2021b)	HuggingFace	45,000	92M
	`pile-of-law/pile-of-law` (Henderson et al., 2022)	HuggingFace		

† Finance data is used exclusively for domain expansion in RQ2.
‡ Law data is tokenized into 2,048-token sequences and used for probe training and evaluation.

Table 15: Architectural comparison of HUDA-MoE-Lite and Full HUDA-MoE. Both share the same Universal Expert, Asymmetric Depth Schedule, domain probe, and training objective.

Component	HUDA-MoE-Lite	Full HUDA-MoE
Experts per domain bank ($N_{\max}$)	1	2
Level-2 token router	—	$\mathbf{W}_{\text{gate}}^{(l,k)}$
Trainable LoRA params	≈27M	≈54M
Level-2 routing equation	N/A	Eq. (5)
Fused forward pass	Eq. (8)	Eq. (9)
Forward pass pseudocode	Alg. 2	Alg. 1
Within-bank routing entropy target	Single expert	$0.10 < H < 0.60$
Shared components		
Universal Expert (frozen backbone)	✓	
Asymmetric Depth Schedule	Eq. (1)	
Domain probe f_ϕ	Eq. (3), 2.1M params, Layer-0 only	
Level-1 routing	Eq. (4), $\tau = 0.05$, $T = 10$	
Training objective	Eq. (6)	
LoRA config	$r = 16$, $\alpha = 32$, dropout$= 0.05$	

LoRA heads + flat router. The *shared domain heads* are combined with a flat token-level router (trained for 500 steps on the mixed corpus). Because this configuration uses the same head checkpoints as HUDA-MoE-Lite and differs only in the router (flat token-level versus sequence-level Level-1), the HUDA-MoE-Lite gap isolates the contribution of sequence-level routing with head training held fixed.

Flat MoE-8. Eight LoRA heads trained *jointly* with a flat token-level router, parameter-matched to full HUDA-MoE (≈54M trainable parameters). As HUDA-MoE also has two experts per bank, this baseline holds the trainable-parameter budget fixed and isolates the two-level routing hierarchy.

Oracle routing. The *shared domain heads* are routed by ground-truth domain labels instead of the learned probe. Upper bound that isolates the cost of label-free routing: the performance gap between HUDA-MoE and Oracle routing comes exclusively from using probe-based instead of oracle bank selection, because the number of heads and the routing granularity are otherwise the same.

HUDA-MoE-Lite. The *shared domain heads* routed by the domain probe (Eq. 3; τ=0.05, T=10), one expert per bank. Implements Algorithm 2 (Eq. 8).

HUDA-MoE. Two LoRA experts per domain bank (N=2) with a learned Level-2 token router within each active bank. It implements Algorithm 1 (Eq. 9). Expert 1 is the corresponding *shared domain head*; Expert 2 is a fresh LoRA head trained

Algorithm 2 HUDA-MoE-Lite Forward Pass

Require: Token embeddings $\mathbf{X} \in \mathbb{R}^{n \times d}$, frozen backbone $\{\mathbf{W}_0^{(l)}\}_{l=1}^{L}$, LoRA experts $\{(\mathbf{A}_k^{(l)}, \mathbf{B}_k^{(l)})\}$, domain probe f_ϕ, threshold τ
Ensure: Output hidden states $\mathbf{H}^{(L)} \in \mathbb{R}^{n \times d}$ **Level-1: Sequence-level domain routing (run once)**
$\bar{\mathbf{h}}^{(0)} \leftarrow \frac{1}{n}\sum_{t=1}^{n} \mathbf{X}[t,:]$ ▷ Mean-pool Layer-0 embeddings
$\mathbf{p} \leftarrow f_\phi\left(\bar{\mathbf{h}}^{(0)}\right)$ ▷ Probe output: $\mathbf{p} \in \Delta^{K-1}$
$\mathcal{B} \leftarrow \{k : p_k > \tau\}$ ▷ Active domain banks
// Layer-wise forward pass
for $l = 1$ **to** L **do**
  $\mathbf{X} \leftarrow \mathrm{LN}\left(\mathbf{X} + \mathrm{MHSA}^{(l)}(\mathbf{X})\right)$ ▷ Frozen self-attention + LayerNorm
  $\Delta\mathbf{X} \leftarrow \mathbf{0}$
  if $l > \lfloor L/2 \rfloor$ **then** ▷ Adapted layers only
    for $k \in \mathcal{B}$ **do**
      $\Delta\mathbf{X} \leftarrow \Delta\mathbf{X} + p_k \cdot \frac{\alpha}{r}\, \mathbf{X}\, \mathbf{A}_k^{(l)\top} \mathbf{B}_k^{(l)\top}$ ▷ Soft LoRA mixture (Eq. 8)
    end for
  end if
  $\mathbf{X} \leftarrow \mathbf{X} + \mathrm{FFN}^{(l)}(\mathbf{X}) + \Delta\mathbf{X}$ ▷ Frozen FFN + LoRA delta
end for
return $\mathbf{X}$

with $\mathrm{lr} = 3 \times 10^{-4}$ and a different data-shuffle seed to encourage expert divergence. The difference between HUDA-MoE-Lite HUDA-MoE captures the effect of Level-2 token routing *along with* the inclusion of a second expert per bank. This additional capacity is regulated by the parameter-matched Flat MoE-8 baseline described above.

Baseline separation. Primary (controlled) baselines share the same LLaMA-3.1-8B backbone, domain training corpus, and evaluation protocol:

- Frozen LLaMA-3.1-8B (no adaptation),
- Flat MoE-4 (jointly trained, ≈27M trainable parameters),
- LoRA heads with a flat token router (≈27M trainable parameters),
- Flat MoE-8 (parameter-matched to HUDA-MoE, ≈54M trainable parameters),
- Oracle routing (upper bound using true domain labels).

Reference-only (uncontrolled) baselines are externally pretrained MoE systems and *should not be interpreted as controlled comparisons*:

- OLMoE-1B-7B (Muennighoff et al., 2025) (7B total, 1B active, 64 experts, top-8 routing), https://huggingface.co/allenai/OLMoE-1B-7B-0924
- DeepSeekMoE-16B (Dai et al., 2024) (16.4B total, 2.8B active, 64 experts, top-6 routing), https://huggingface.co/deepseek-ai/deepseek-moe-16b-base
- Mixtral-8x7B (Jiang et al., 2024) (47B total, 13B active, 8 experts, top-2 routing), https://huggingface.co/mistralai/Mixtral-8x7B-v0.1.

Backbone LLMs are:

- LLaMA-3.1-8B (Grattafiori et al., 2024), https://huggingface.co/meta-llama/Llama-3.1-8B
- Qwen-2.5-7B (Yang et al., 2024), https://huggingface.co/Qwen/Qwen2.5-7B
- Mistral-7B (Jiang et al., 2024), https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3.

E Benchmark Dataset

The benchmark datasets we use in our paper: **MBPP** (Austin et al., 2021). Mostly Basic Programming Problems: 3-shot, pass@1, n=500. Tests code generation capability, evaluated by executing generated code against unit tests.

MedQA-USMLE (Jin et al., 2021). US Medical Licensing Examination: 0-shot, 4-way multiple-choice accuracy, n=1,273. Tests clinical biomedical reasoning.

PubMedQA (Jin et al., 2019). Biomedical research question answering: 0-shot, 3-way classification (yes/no/maybe), n=500. Tests biomedical reading comprehension over PubMed abstracts; complements MedQA by targeting literature-grounded inference rather than clinical decision-making.

GSM8K (Cobbe et al., 2021). Grade School Math: 5-shot, exact match (strict), n=1,319. Tests mathematical reasoning via multi-step word problems.

MMLU (Hendrycks et al., 2021a). Massive Multitask Language Understanding: 5-shot accuracy, 57 subjects, n=14,042. Tests cross-domain general knowledge and serves as the non-forgetting probe: all configurations should match the frozen baseline within measurement noise.

Table 16: Routing distributions (%) for the original probe. The probe routes most benchmark examples to the expected domain bank, while mixed-domain benchmarks such as MMLU naturally distribute across multiple experts.

Benchmark	Expected	Code	Biomed	Math	News	Correct?
MBPP	Code	**91.0**	0.5	8.0	0.5	✓
MedQA	Biomed	8.5	**74.0**	6.5	11.0	✓
GSM8K	Math	9.5	0.5	**86.0**	4.0	✓
MMLU	Mixed	18.0	21.5	16.5	**44.0**	—
MATH	Math	12.0	0.5	**86.5**	1.0	✓
PubMedQA	Biomed	10.5	**78.0**	4.5	7.0	✓

Table 17: Asymmetric Depth Schedule validated across three backbone LLMs. In-domain and OOD Δ are mean relative PPL changes across four domains. BPB enables fair cross-tokenizer comparison.

Backbone	Adapted Layers	Avg In-Domain Δ	Avg OOD Δ (ex. Biomed)
LLaMA-3.1-8B	17–32	-9.6%	$+2.5\%$
Mistral-7B-v0.3	16–31	-6.3%	$+2.7\%$
Qwen-2.5-7B	15–27	-5.8%	$+2.0\%$

F Implementation Details

LegalBench-tuned LoRA head: `ASethi04/meta-llama-Llama-3.1-8B-legalbench-1-lora-1-0.004`.

Hardware. Domain probe experiments run on a single NVIDIA A100-SXM4-40GB. LoRA head training uses up to five A100-40GB instances in parallel via Modal cloud infrastructure. Downstream benchmark evaluation and full HUDA-MoE training use a single NVIDIA H100-80GB. In total, it costs us about 445 GPU hours for all the experiments.

Software. PyTorch, HuggingFace Transformers, PEFT for LoRA management, BITSANDBYTES for NF4 quantization, and LM-EVALUATION-HARNESS for standardized benchmark evaluation.

Artifact licenses and terms of use. All scientific artifacts used in this work are publicly available and were used for non-commercial research purposes consistent with their respective licenses and intended use. The LLaMA-3.1-8B backbone is used under the Llama 3.1 Community License, and Mistral-7B and Qwen-2.5-7B are used under the Apache-2.0 license. The external MoE systems included as reference points (OLMoE-1B-7B, DeepSeekMoE-16B, Mixtral-8x7B) and the external LegalBench-tuned LoRA adapter are obtained from public repositories and used under the licenses specified by their releases. The training and evaluation datasets listed in Table 14 are released under their respective open licenses (e.g., Apache-2.0, MIT, and Creative Commons variants as specified by each source); the legal corpora (CUAD, Pile of Law) are used in accordance with the research-use terms stated by their maintainers. We do not redistribute any third-party artifacts. The LoRA expert heads and probes trained in this work are derivatives of their respective backbones and, if released, will be made available under terms compatible with those backbones' licenses, for research use only.

G Domain Probe Quality

The HUDA-MoE probe achieves ARI = 0.937 against true domain labels, which is slightly higher than the Layer-32 K-Means++ reference from which it was distilled (0.926) and all representation baselines. These baselines include Sentence-BERT (0.674) and direct Layer-0 K-Means++ (0.665), as shown in Table 18. The MLP learns smoother decision boundaries than nearest-centroid assignment, and the ARI variance across five seeds is 9×10^{-6}, confirming highly stable probe quality across seeds. True domain labels are used only to compute evaluation metrics; they are not used during probe training or checkpoint selection.

Routing Distribution Table 16 shows the average routing percentage on three LLM-backbones using HUDA-MoE. It yields three findings: **(1)** Domain-specific benchmarks route to the expected experts: MBPP to Code (91.0%), MedQA and PubMedQA to Biomed (74.0% and 78.0%). **(2)** Math routing is reliable but not perfectly isolated: GSM8K and MATH route to Math at 86.0% and 86.5%, with residual routing to Code and News in boundary cases. **(3)** Mixed-domain benchmarks naturally produce mixed routing: MMLU dis-

Table 18: Domain probe validation on 200K sequences. Probe quality vs. representation baselines. Best result is **bold**.

Representation	ARI	AMI	Mapped Acc.
Random assignment	0.000	0.000	25.0%
TF-IDF + K-Means++	0.807	—	—
S-BERT + K-Means++	0.674	—	—
Layer 0 K-Means++	0.665	0.711	82.5%
Layer 32 K-Means++	0.926	0.902	97.1%
Oracle MLP	∼0.990	∼0.990	∼100%
HUDA-MoE probe	**0.937**	**0.913**	**97.6%**

tributes across all four banks, reflecting its heterogeneous subject coverage.

Probe Distillation The HUDA-MoE probe achieves ARI = 0.937, exceeding the Layer-32 K-Means++ clustering (ARI = 0.926) that generated its training pseudo-labels. A classifier trained on noisy labels cannot correct those labels in general, so this result requires explanation.

The key is the difference between hard and soft assignments. K-Means++ assigns each sequence to its *nearest centroid* via a hard Voronoi boundary, regardless of how close that sequence lies to the boundary. The probe, by contrast, learns a *soft decision surface* that can differ from the nearest-centroid rule wherever the Voronoi boundary passes through high-density regions. Near the Math-Code and Math-News boundaries, K-Means generates most of its 5.6% error. The MLP learns that these sequences have genuinely mixed vocabulary statistics and outputs a probabilistic assignment rather than forcing a hard cluster label. This generalization is consistent with the routing behavior observed in Table 6. Math sequences activate 1.96 banks on average, reflecting the same boundary uncertainty the MLP learned to represent during probe training. The result demonstrates that cross-layer distillation does not merely compress the K-Means signal.

H Multi-Architecture Depth Schedule Results

Table 17 reports in-domain and out-of-distribution (OOD) perplexity changes for the Asymmetric Depth Schedule across three backbone LLMs.

I Domain Probe Confusion Matrix

Tables 19 report the full confusion analysis of the domain probe on the 40K held-out test set (10K sequences per domain).

Table 19: Per-class precision, recall, and F1-score.

Domain	Precision	Recall	F1	Support
Code	0.9754	0.9866	0.9810	9,998
Biomed	0.9998	0.9883	0.9940	10,003
Math	0.9620	0.9438	0.9528	10,033
News	0.9657	0.9841	0.9749	9,966

(1) Biomed is highly distinctive (F1 = 0.994): precision is 99.98%, with only 1.17% of sequences misclassified, all as Math, and none as Code or News. **(2) Math is the most challenging domain** (F1 = 0.953), with confusion mainly along two axes: Math→Code (2.17%), likely involving algorithms or pseudocode, and Math→News (3.45%), likely expository prose whose token statistics resemble news. **(3) Code–Math boundary**: 132 Code sequences (1.32%) are classified as Math, mirroring the Math→Code confusion. **(4) News is moderately confused with Math** (1.25%), likely when scientific or financial articles contain numerical content resembling mathematical text.