DoMoE: Domain-Aware Semantic Expert Prediction for Efficient MoE Inference Under Expert Offloading

[1]Yao Mu , [2]Fahao Chen , [1]Wenbin Zhu , [1]Mengying Zhao , [1]Zhaoyan Shen* , [1]Dongxiao Yu
[1]School of Computer Science and Technology, Shandong University, Qingdao, China
[2]School of Artificial Intelligence, Shandong University, Jinan, China
{yaomu, wenbinzhu}@mail.sdu.edu.cn, fhchen@ieee.org,{shenzhaoyan, dxyu}@sdu.edu.cn

Abstract

Mixture-of-Experts (MoE) large language models improve inference efficiency through sparse expert activation, but deployment on resource-constrained devices remains challenging due to the large expert parameter footprint. Expert offloading mitigates this issue by loading experts on demand, yet its effectiveness critically depends on accurate and efficient expert prediction: inaccurate predictions incur redundant expert transfers, while overly expensive predictors negate latency benefits. Existing trajectory-based methods suffer from low accuracy, whereas semantic-based approaches incur prohibitive similarity-matching overhead. We propose DoMoE, a domain-aware MoE inference system that exploits domain locality in inference workloads. DoMoE organizes routing information into domain-specific expert routing tables, restricts semantic matching to domain-relevant tokens and explicitly balances prediction accuracy against prediction overhead. Experiments show that DoMoE achieves a 1.32× average throughput improvement and a 1.22× increase in expert hit ratio across multiple MoE models and workloads, enabling efficient and accurate expert routing for practical inference.

1 Introduction

Mixture-of-Experts (MoE)-based large language models (LLMs) have recently demonstrated strong performance across a wide range of downstream tasks [Maity and Saikia, 2025; Xu *et al.*, 2025; Fieberg *et al.*, 2024]. An MoE-based LLM consists of multiple specialized expert networks, aiming to capture different inputs. By maintaining a large pool of expert parameters while activating only a small subset during inference, MoE models improve predictive accuracy and reduce computational cost [Masoudnia and Ebrahimpour, 2014]. However, their substantial overall parameter footprint poses significant challenges for deployment [Hadia *et al.*, 2025], particularly on resource-constrained devices [Zhang *et al.*, 2025]. Existing model compression techniques (e.g., quantization [Hu *et al.*, 2025a] and pruning [Lu *et al.*, 2024]) and MoE-specific parameter reduction approaches (e.g., expert merging [He *et al.*, 2023]) can mitigate this issue by shrinking model size, but they often degrade inference accuracy [Liu *et al.*, 2025].

To enable efficient MoE deployment without sacrificing accuracy, expert offloading [Hu *et al.*, 2025b] has emerged as a dominant strategy. By exploiting the sparse activation property of MoE models, expert offloading dynamically loads only the activated experts onto computation units (e.g., GPU memory), while keeping inactive experts in lower-cost space (e.g., host memory). A key challenge in expert offloading is accurately predicting which experts will be activated, as mispredictions can introduce substantial expert loading overhead and significantly diminish performance gains.

Existing studies propose trajectory-based and semantic-based methods for expert activation prediction. Trajectory-based methods [Xue *et al.*, 2024] predict expert activations by matching partial activation paths observed in earlier layers against historical expert trajectories, leveraging inter-layer correlations to prefetch experts with minimal prediction overhead. Although computationally efficient, these methods suffer from low prediction accuracy due to insufficient information available in early MoE layers, which further propagates prediction errors to subsequent layers. In contrast, semantic-based approaches [Yu *et al.*, 2025] predict expert activations using token-level semantic similarity, comparing current token embeddings with historical representations to identify experts associated with similar inputs. While these methods achieve higher prediction accuracy by exploiting richer semantic information, computing similarity against all historical tokens incurs significant computational overhead.

In this work, we propose DoMoE, an efficient MoE inference system with a novel expert prediction mechanism to mitigate runtime I/O overhead. DoMoE follows the semantic-based prediction paradigm but introduces a domain-aware design to substantially improve prediction efficiency. Our key observation is that token retrieval for expert prediction is inherently domain-dependent: tokens that exhibit similar expert selection behaviors typically come from the same domain (e.g., healthcare, finance, or law), rather than being uniformly distributed across the entire vocabulary space. The rationale is that MoE gating functions and experts are implicitly specialized under domain-specific data distributions during training, making expert selection patterns more trans-

*Corresponding author.

ferable within the same domain. Therefore, it is unnecessary to compute semantic similarity between a target token and all historical tokens. Instead, DoMoE restricts similarity search to historical tokens within the same domain as the target token, significantly reducing the search space while preserving prediction quality.

Based on this insight, DoMoE performs expert prediction with an offline expert routing table construction phase and an online expert identification phase. Offline, DoMoE constructs multiple domain-specific *expert routing tables*, where each table stores historically generated tokens from a particular domain together with their corresponding expert selections. Online, when an inference request arrives, DoMoE first matches the domain of the input query and loads the corresponding *expert routing table* for expert prediction. Note that expert prediction is only applied during the decoding phase, since the prefill phase typically requires loading all experts to process the query with multiple tokens in parallel. Specifically, for each generated decoding token, DoMoE computes similarity between its embedding and the embeddings of tokens in the selected *expert routing table* to retrieve the most similar historical token. DoMoE then prefetches the experts selected by the retrieved token into GPU memory before executing subsequent layers. Any experts that are not successfully prefetched are loaded on demand at runtime, enabling expert offloading with minimal impact on end-to-end inference latency.

To make DoMoE practical and efficient, we must address two key challenges. First, efficiently matching the domain of an incoming query and loading the corresponding *expert routing table* is non-trivial. Therefore, DoMoE requires a lightweight domain matching method with minimal interference to end to end inference execution. Second, token retrieval remains a non-negligible bottleneck even when search is restricted to a domain-specific routing table. Existing offloading systems typically perform layer-by-layer prediction, using the current-layer token embedding to predict experts for the next layer and overlapping expert loading with ongoing computation. However, invoking semantic matching at every layer incurs substantial cumulative overhead. A natural alternative is multi-layer-ahead prediction, which amortizes retrieval cost by predicting experts for multiple upcoming layers at once. Nevertheless, extending the prediction horizon reduces accuracy, leading to more expert misses and redundant prefetching that ultimately offset the latency savings. This trade-off motivates the need for an adaptive strategy that jointly optimizes prediction frequency and prediction quality.

To address the above challenges, DoMoE incorporates the following two designs.

Domain-aware expert routing table matching mechanism. DoMoE precomputes compact prototype embeddings for each domain-specific expert routing table. During the prefill phase, DoMoE clusters token embeddings from the input query and matches the resulting cluster representations against these prototypes to select the most relevant domain expert routing tables.

Adaptive expert prediction strategy . Instead of predicting expert activations only for the next layer, DoMoE predicts across multiple layers ahead. To balance prediction horizon and accuracy, we use a sampling-based statistical method to select an optimal set of prediction layers, explicitly accounting for both prediction accuracy and the overhead incurred by expert loading. This design ensures that the cost of prediction does not outweigh the latency savings enabled by more efficient expert prediction.

We evaluate DoMoE against representative expert offloading baselines across multiple MoE models and domain workloads. DoMoE consistently achieves the highest end-to-end inference throughput while maintaining superior expert hit ratios. Compared with the state-of-the-art method, DoMoE delivers a $1.32\times$ average throughput improvement and a $1.22\times$ increase in expert hit ratio by jointly improving expert routing accuracy and execution efficiency. Ablation results further show that domain-aware expert routing tables provide the dominant performance gains, while adaptive expert prediction contributes additional throughput improvements despite a slight reduction in hit ratio. Overall, these results demonstrate that DoMoE effectively balances routing accuracy and prediction overhead, yielding robust system-level performance gains.

The main contributions of the work are summarized as follows:

- We introduce a prototype-based routing mechanism that selects the top-2 domain-specific expert routing tables by matching prefill token embeddings against precomputed routing prototypes, enabling robust domain-aware expert prediction without explicit domain labels.
- We address the accuracy–latency trade-off in expert activation prediction by introducing a prediction optimization strategy that balances prediction cost against prefetching benefits, preventing overly expensive predictions from reducing overall inference throughput.
- Experimental results show that DoMoE outperforms the state-of-the-art method, achieving a $1.32\times$ improvement in overall throughput and a $1.22\times$ increase in expert hit ratio.

2 Background and Motivation

2.1 MoE and MoE Based LLM Serving

Mixture-of-Experts (MoE) is an efficient architecture for large language models (LLMs) that introduces multiple specialized *experts*, where each expert is a feed-forward network responsible for processing a subset of input tokens, typically associated with specific domains or tasks [Xu *et al.*, 2026]. MoE models[1] improve inference efficiency by activating only a small subset of experts for each input, in contrast to dense LLMs that execute all parameters during inference [Yuksel *et al.*, 2012]. By dynamically routing tokens to a limited set of relevant experts, MoE models reduce computation and memory access while retaining strong model capacity and flexibility.

[1]In this manuscript, MoE models denote LLMs deployed with the MoE architecture.

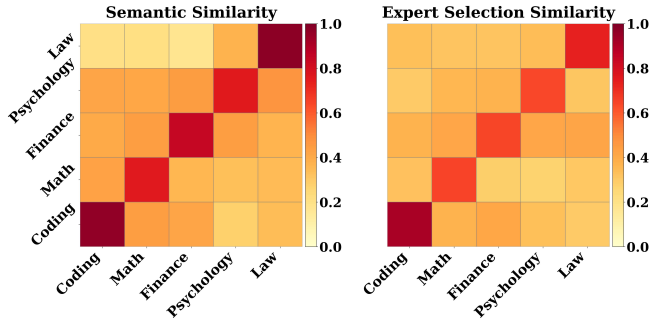

Figure 1: Cross-domain similarity analysis for five specialized datasets

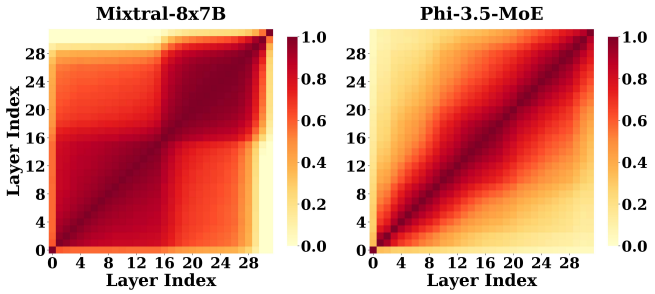

Figure 2: Layer-to-layer embedding similarity matrix for Mixtral-8X7B and Phi-3.5-MoE models.

While MoE models are efficient at inference time, deploying them, especially on resource-constrained edge devices, remains challenging because their large parameter footprints often exceed the limited GPU memory available on devices [Kong *et al.*, 2025]. Existing approaches for enabling MoE deployment on edge devices can be broadly categorized into compression-based and offloading-based techniques. Compression-based methods, including quantization, parameter pruning, and gating optimization [Kim *et al.*, 2023; Yang *et al.*, 2024; Hwang *et al.*, 2024], reduce resource consumption at the cost of model fidelity. However, quantization lowers numerical precision and may degrade accuracy [Li *et al.*, 2024], while pruning methods drop experts, potentially weakening expert specialization and reducing robustness [Ma *et al.*, 2025]. Although optimizing the gating function can further improve efficiency, it may also alter routing behaviors and harm generalization across diverse inputs and domains [Zhong *et al.*, 2024]. In contrast, offloading-based techniques such as expert offloading preserve model accuracy by retaining the full MoE architecture, but they introduce additional I/O overhead that can significantly increase inference latency [Yang *et al.*, 2025]. Therefore, achieving high inference efficiency without sacrificing accuracy remains a central challenge for practical MoE deployment on personal devices.

2.2 Expert Prediction Methods

To mitigate the I/O overhead of expert loading, prior works propose predicting expert activations so that the required experts can be prefetched in advance. One representative line of work adopts trajectory-based prediction [Xue *et al.*, 2024]. By exploiting sparse expert activation and skewed reuse patterns, these methods track expert activation trajectories in early layers to guide expert caching and prefetching. However, expert activations in early layers are often only weakly correlated with routing decisions in later layers, leading to low prediction accuracy. As a result, frequent mispredictions trigger unnecessary expert transfers and reduce overall inference efficiency.

Another line of work focuses on semantic-based methods to improve expert prediction accuracy [Yu *et al.*, 2025]. These methods maintain expert probability maps and semantic embeddings, and predict future expert activations via semantic similarity and expert-path matching. Although this approach can substantially improve expert cache hit rates, it introduces significant memory and computational overhead due to large-scale embedding storage and expensive similarity search, which limits scalability and practical efficiency.

2.3 Motivation

Domain locality in expert activation. In practice, inference is often domain-oriented, where each domain exhibits relatively stable topical focus and recurring domain-specific terminology. As a result, queries and responses within the same domain tend to concentrate on a characteristic subset of the vocabulary, leading to a skewed token distribution with high-frequency domain-relevant tokens. Figure 1 empirically shows that tokens within the same domain not only form a more coherent semantic neighborhood in embedding space, but also exhibit more transferable expert selection behaviors under MoE routing. In contrast, when tokens originate from different domains, the retrieved semantic neighbors are less informative for expert prediction, and the associated routing patterns become less consistent, suggesting that expert activation behavior is strongly conditioned on domain context rather than being globally transferable across all tokens.

Motivated by this observation, we design *domain-specific expert routing tables* to guide expert activation prediction. Instead of maintaining a single global table that aggregates tokens from all domains, our approach partitions routing information by domain and retains only the most representative, high-frequency tokens within each domain. This domain-aware organization yields compact information for expert activation prediction, which substantially reduces storage overhead and narrows the search space during semantic matching. Consequently, expert retrieval latency can be reduced while preserving informative routing signals, enabling efficient and accurate expert prediction for MoE inference.

Inter-layer similarity of token embeddings. As shown in Figure 2, embeddings of the same token exhibit consistently high similarity between adjacent layers, with an average cosine similarity of 0.96 and peak values exceeding 0.99 across 32 layers. This strong locality indicates that token representations change smoothly across nearby layers, suggesting that expert activation behavior is also likely to remain stable within a short layer range.

This observation implies that performing expert prediction independently at every layer is largely redundant: since successive layers operate on highly similar token embeddings, a

single prediction can reliably guide expert selection for multiple subsequent layers with minimal loss in accuracy. However, we also observe that embedding similarity degrades as the layer distance increases, indicating that representations diverge more substantially across distant layers. Consequently, predicting too far ahead may lead to inaccurate expert selection and reduced hit rates.

These findings provide a principled basis for our adaptive expert prediction strategy, which leverages high inter-layer similarity to amortize prediction overhead by covering multiple layers per prediction, while carefully limiting prediction range to avoid accuracy degradation.

3 Methodology

3.1 DoMoE Overview

In this paper, we introduce DoMoE, a domain-aware MoE inference system that accelerates expert offloading by predicting expert activations using domain-specific *expert routing tables*. DoMoE adopts a two-phase design that decouples offline table construction from online inference, enabling accurate expert routing with minimal runtime overhead. As shown in Figure 3, during the *offline phase*, DoMoE organizes historical tokens into domain-specific routing tables and records their embeddings together with expert activation paths. It further computes prototype representations for each domain table via clustering, which will be used for efficient domain identification at runtime. During the *online phase*, DoMoE performs lightweight domain matching during prefill stage to select the most relevant domain tables, and then integrates expert prediction and prefetching into decoding stage to proactively prepare expert parameters before execution.

System Workflow

- **Domain partitioner:** Builds domain-specific expert routing tables offline by grouping historical tokens according to domain categories (e.g., Law, Finance, and Math). For each domain table, it stores token embeddings together with expert activation paths, and computes a small set of clustering-based prototypes to summarize the domain's embedding distribution.
- **Domain matching:** Invoked during prefill stage to infer the domain context of an incoming query. It clusters the query token embeddings to obtain a few representative centers and matches them against domain prototypes, selecting the top-2 most relevant domain tables for subsequent expert prediction.
- **Expert predictor:** During decoding stage, it retrieves semantically similar historical tokens from the selected domain tables and predicts future expert activations by aggregating their expert activation paths using similarity-weighted voting to generate a ranked list of candidate experts.
- **Expert prefetcher:** Prefetches the predicted expert parameters from CPU memory to GPU memory ahead of execution, reducing expert-loading stalls on the critical inference path. Any missed experts are fetched on demand to ensure correctness.

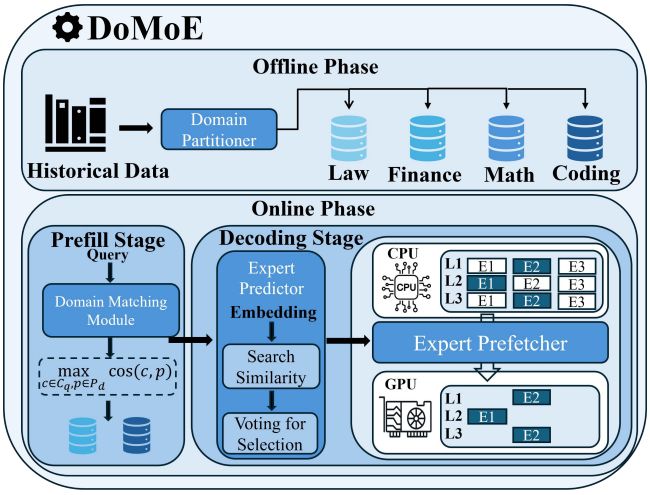

Figure 3: Flowchart of DoMoe's architecture and work flow

3.2 Domain-Aware Expert Routing Table Matching Mechanism

To enable domain-aware expert routing under dynamic inference execution, we propose a *domain-aware expert routing table matching mechanism* that efficiently identifies a small set of relevant expert routing tables while remaining robust to domain ambiguity.

Prototype construction. For each domain-specific expert routing table, DoMoE precomputes a compact set of *routing prototypes* by clustering historical token embeddings stored in that table. Each prototype corresponds to a representative semantic cluster center that summarizes the dominant token distribution of the domain. This construction is performed offline, and the resulting prototypes are reused during inference to enable fast domain matching with negligible runtime overhead.

To keep prototypes consistent with evolving table contents, DoMoE supports incremental prototype maintenance when new tokens are appended to an expert routing table. In practice, this update can be amortized and executed asynchronously (e.g., periodically or in the background), ensuring that prototype freshness is maintained without introducing overhead on the critical inference path.

Domain matching. When a new query arrives, DoMoE performs domain matching during the prefill phase. We extract token embeddings from prefill computation and obtain a compact query representation by clustering these embeddings and selecting their cluster centers. Let C_q denote the resulting set of query cluster centers. For each domain d with routing prototypes P_d, we compute a matching score:

$$s(d) = \max_{c \in C_q,\, p \in P_d} \cos(c, p), \tag{1}$$

where the max operator captures the strongest semantic evidence between the query and the domain table. Based on these scores, DoMoE selects the *top-2* domain-specific expert routing tables for subsequent expert prediction. This bounded selection ensures that decoding-time retrieval remains efficient while preserving domain locality.

Domain matching may be ambiguous, and a token could span multiple domains. To avoid error from single-table decisions, DoMoE adopts a confidence-aware selection strategy: when the similarity margin between the top-ranked domains is small, both candidate tables are jointly consulted rather than committing to a single domain. This design avoids reliance on a global fallback table while still tightly bounding the routing search space.

3.3 Adaptive Expert Prediction Strategy

Prior works on expert offloading typically invoke expert prediction at each MoE layer to maximize routing accuracy. While this design can achieve high expert prediction accuracy, it introduces substantial runtime overhead, especially for semantic-based predictors that require expensive similarity search. As a result, the increased prediction cost may negate the latency savings from reduced expert loading stalls, limiting end-to-end inference efficiency.

A more effective strategy is to issue expert predictions less frequently and predict experts for multiple subsequent layers. However, this introduces a fundamental trade-off: increasing the prediction horizon (i.e., predicting farther ahead) reduces the number of prediction invocations and thus lowers prediction overhead, but it also tends to decrease prediction accuracy and incur more expert misses and redundant prefetching. Conversely, short-horizon prediction improves accuracy but requires more frequent prediction, leading to higher cumulative overhead. Therefore, our objective is to balance prediction accuracy and prediction cost to minimize end-to-end inference latency. This requires determining (i) *which layers* should trigger expert prediction and (ii) *how many future layers* each prediction should cover.

Problem formulation. We consider an MoE model with L sequential layers indexed by $\mathcal{L} = \{1, \dots, L\}$. Let $x_l \in \{0, 1\}$ indicate whether expert prediction is invoked at layer l.

Expert prediction at a given layer affects not only that layer but also all subsequent layers until the next prediction is performed. To capture this dependency, we first define, for each layer l, its distance to the most recent prediction layer. Specifically, we introduce the prediction offset function $d(l)$, defined as

$$d(l) = l - \max\{l' \mid l' \leq l \text{ and } x_{l'} = 1\}. \tag{2}$$

Here, $d(l)$ represents the number of layers elapsed since the last prediction invocation, and therefore determines how stale the prediction becomes at layer l.

Using this offset, we model the system cost at each layer as a function of prediction frequency and prediction staleness. Let c^{pred} denote the latency overhead of invoking expert prediction. Let $\hat{p}_{d(l)}$ denote the empirically estimated misprediction probability at offset $d(l)$ layers after a prediction, and let c^{load} be the expected expert-loading latency. The expected end-to-end inference latency can then be expressed as

$$\min_{\{x_l\}} \sum_{l=1}^{L} \underbrace{c^{\text{pred}} x_l}_{\text{prediction overhead}} + \underbrace{\hat{p}_{d(l)}\, c^{\text{load}}}_{\text{expected misprediction penalty}} . \tag{3}$$

Model	Params (Act. / Total)	Experts (Act. / Total)	Layers
Mixtral-8×7B	12.9 B / 46.7 B	2 / 8	32
Phi-3.5-MoE	6.6 B / 42.0 B	2 / 16	32
Qwen1.5-MoE	2.7 B / 14.3 B	4 / 60	24
Qwen3-30B-A3B	3.3 B / 30.5 B	8 / 128	48

Table 1: Characteristics of the MoE models used in our experiments.

This formulation explicitly captures the trade-off between prediction frequency and prediction horizon: invoking prediction more frequently reduces the offset $d(l)$ and lowers misprediction cost, but increases cumulative prediction overhead, while less frequent prediction amortizes prediction cost at the expense of higher misprediction penalties. Solving this optimization yields the theoretical optimal set of prediction layers and the corresponding optimal number of prediction invocations. Although solving this integer program online is impractical, it motivates the multi-layer prediction strategy adopted in DoMoE.

4 Evaluation

Environment. All experiments are conducted on a single server equipped with an NVIDIA L40 GPU, an Intel Xeon Platinum 8480+ CPU, and 1 TB of system memory.

Models. We evaluate our system on three representative MoE models with diverse architectural characteristics, as summarized in Table 1. (1) Mixtral-8×7B [Jiang *et al.*, 2024] adopts a sparse MoE design with 8 experts per layer and activates the top-2 experts per token during inference. (2) Phi-3.5-MoE [Abdin *et al.*, 2024] contains 16 experts per layer and similarly activates the top-2 experts per token, providing larger expert capacity under comparable activation sparsity. (3) Qwen1.5-MoE [Team, 2024] features a substantially larger expert pool with 60 experts per layer and activates the top-4 experts per token, resulting in higher routing complexity and greater expert loading pressure. (4)Qwen3-30B-A3B [Team, 2025] is a MoE model with 30.5B total parameters and 3.3B activated parameters, using 48 layers, 128 experts, and 8 activated experts per token to achieve efficient inference. Together, these models span a broad range of expert counts, sparsity levels, and parameter scales, enabling systematic evaluation of DoMoE under diverse MoE inference settings.

Datasets. We evaluate all methods on five publicly available datasets covering diverse domains and interaction patterns, including multi-turn dialogue, intent-oriented conversations, and reasoning-centric traces. (1) DISC-Law [Yue *et al.*, 2024] focuses on legal-domain interactions, spanning tasks such as legal information extraction, judgment prediction, document summarization, and question answering. (2) PsyDTCorpus [Xie *et al.*, 2025] contains multi-turn counseling dialogues derived from a specific psychotherapist; it includes 5,000 high-quality conversations synthesized from 5,000 single-turn samples using a digital-twin approach to capture therapist style and therapeutic patterns. (3) Banking77 [Casanueva *et al.*, 2020] is a banking-domain intent classification benchmark with 77 intent categories, which we convert into conversational-style evaluation prompts for MoE

Model	Method	Throughput (Tokens/s)			Expert Hit Ratio			CPU Footprint(MB)
		DISC-Law	Banking77	Mixed	DISC-Law	Banking77	Mixed	
Mixtral-8×7B	MoE-Infinity	0.65	0.79	0.60	46%	48%	43%	–
	FMoE	0.92	1.10	0.85	62%	59%	52%	25.2
	DoMoE	**1.24**	**1.43**	**1.21**	**69%**	**72%**	**66%**	**25.2**
Phi-3.5-MoE	MoE-Infinity	1.03	1.08	0.92	50%	50%	38%	–
	FMoE	1.67	1.53	1.34	66%	59%	49%	25.95
	DoMoE	**2.32**	**1.92**	**1.95**	**72%**	**67%**	**68%**	**25.95**
Qwen-1.5-MoE	MoE-Infinity	1.05	1.02	0.89	47%	49%	39%	–
	FMoE	1.48	1.47	1.25	56%	55%	44%	24.21
	DoMoE	**1.83**	**1.71**	**1.68**	**67%**	**65%**	**64%**	**24.21**
Qwen3-30B-A3B	MoE-Infinity	0.77	0.83	0.72	39%	43%	36%	–
	FMoE	0.98	1.15	0.97	52%	57%	50%	25.56
	DoMoE	**1.28**	**1.45**	**1.33**	**68%**	**70%**	**67%**	**25.56**

Table 2: Throughput, expert hit ratio, and CPU footprint comparison across different MoE models and methods. **Mixed** denotes the mixed-domain workload composed of PsyDTCorpus, Agentic Coding CoT, and PRM800K.

inference. (4)Agentic Coding CoT [AlicanKiraz0, 2025] consists of approximately 20 GB of coding-related traces curated from GitHub and processed using Minimax-M2, producing structured tool-usage and chain-of-thought style examples. (5) PRM800K [Lightman *et al.*, 2023] provides 800,000 step-level correctness annotations for model-generated solutions on MATH problems, representing long-form reasoning trajectories with fine-grained supervision signals.

Baselines. We compare DoMoE against two representative expert offloading systems that follow different expert prediction paradigms. (1) MoE-Infinity [Xue *et al.*, 2024] serves as a trajectory-based baseline that predicts expert activations from early-layer routing trajectories to guide expert caching and prefetching. (2) FMoE [Yu *et al.*, 2025] represents a semantic-based baseline that performs expert prediction via semantic similarity matching and expert-path retrieval. Since FMoE does not provide an open-source implementation, we reimplement it by extending the MoE-Infinity codebase and use this implementation for evaluation.

Implementation Details. We implement DoMoE based on MoE-Infinity system [Xue *et al.*, 2024], with approximately 4K LoC to support domain-aware expert routing table construction, table matching, and adaptive expert prediction. For each dataset, we use the first 70% of samples as a profiling split to collect token embeddings and expert activation paths for all methods. DoMoE maintains domain-specific expert routing tables, where each table is built from the profiling split of the corresponding domain. In contrast, baseline methods do not distinguish domains and instead aggregate profiling data from all domains into a single repository. We then evaluate all methods on the remaining 30% of samples. We report results on two single-domain workloads (*Law* and *Banking*) and on a mixed-domain workload constructed by interleaving queries from the other three datasets.

4.1 Overall Performance

Table 2 reports end-to-end inference throughput and expert hit ratio on DISC-Law, Banking77, and a mixed-domain workload. Overall, compared with FMoE, DoMoE achieves an average throughput improvement of 1.32× across all models and workloads, while increasing the expert hit ratio by 1.22×, demonstrating both higher routing accuracy and substantially better end-to-end inference efficiency.

Across all evaluated settings, DoMoE consistently delivers the highest inference throughput. On Mixtral-8×7B, DoMoE improves throughput by approximately 1.9×–2.1× over MoE-Infinity and 1.35×–1.45× over FMoE, with the largest gains observed on the mixed-domain workload. On Phi-3.5, DoMoE achieves up to 2.25× higher throughput than MoE-Infinity and around 1.4× higher throughput than FMoE. Similar trends hold for Qwen1.5-MoE, where DoMoE delivers 1.7×–1.9× higher throughput than MoE-Infinity and 1.25×–1.35× higher throughput than FMoE, despite the substantially larger expert pool and increased routing complexity. These results indicate that DoMoE consistently translates domain-aware routing optimizations into end-to-end throughput gains across diverse MoE architectures.

DoMoE also attains consistently higher expert hit ratios across all datasets and models. Compared with MoE-Infinity, DoMoE improves the hit ratio by approximately 1.2×–1.3×, and further increases it by 1.10×–1.15× over FMoE, particularly under mixed-domain workloads. This improvement stems from DoMoE's domain-aware organization of routing information: unlike FMoE, which stores heterogeneous expert histories in a unified table and thus retains a large fraction of domain-irrelevant entries under the same table size, DoMoE partitions routing information by domain, preserving a higher density of domain-relevant signals and consequently achieving more accurate expert prediction. Importantly, higher hit ratio alone does not guarantee better system performance: although FMoE improves routing accuracy over MoE-Infinity, its throughput gains are limited by the

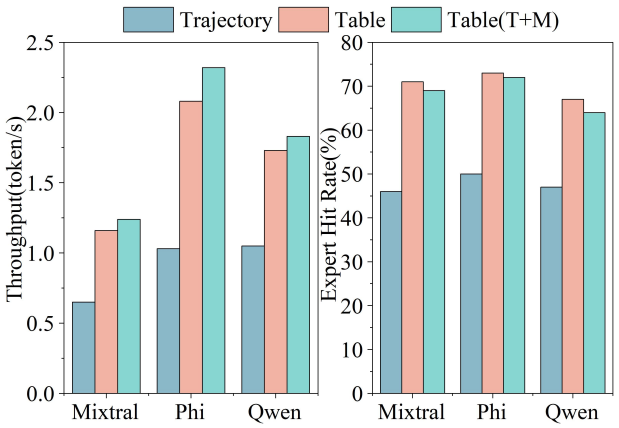

Figure 4: Ablation study of DoMoE

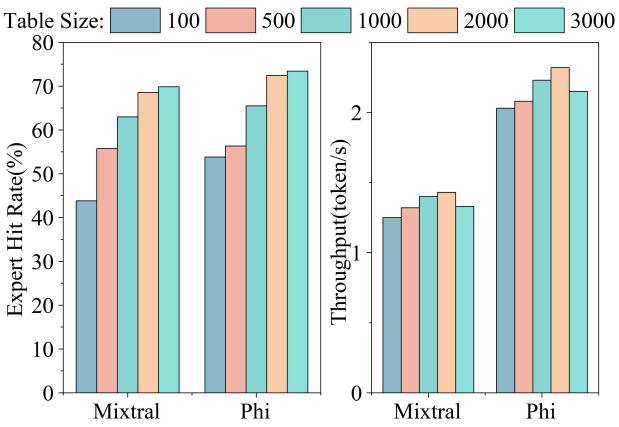

Figure 5: Sensitivity study of DoMoE

high cost of frequent semantic prediction. In contrast, DoMoE maintains comparable or higher hit ratios while achieving substantially higher throughput, confirming that its adaptive expert prediction strategy effectively balance prediction accuracy against prediction overhead.

4.2 Ablation Study

We conduct an ablation study on the mixed dataset to quantify the contribution of DoMoE's key components. We compare three configurations: (1) *Trajectory*, which uses trajectory-based expert prediction; (2) *Table*, which replaces it with domain-aware expert routing tables; and (3) *Table(T+M)*, the full DoMoE system with prediction optimization. Figure 4 summarizes the results. Overall, domain-aware routing tables provide the main performance gains by improving routing quality, while prediction optimization further improves efficiency by reducing unnecessary prediction overhead.

Compared with *Trajectory*, *Table* improves throughput by 1.78× on Mixtral-8×7B (0.65 → 1.16 tokens/s), 2.02× on Phi-3.5 (1.03 → 2.08), and 1.65× on Qwen1.5-MoE (1.05 → 1.73). The expert hit ratio also increases by about 1.25× across models. These gains show that domain-aware routing tables remove task-irrelevant historical routes from the search space, making semantic matching more focused and less costly. As a result, expert prediction becomes both more accurate and more efficient. This confirms that organizing routing data by domain is the dominant factor behind DoMoE's offloading performance.

The *Table(T+M)* configuration further improves throughput over *Table* by 1.07×, 1.12×, and 1.06× on Mixtral-8×7B, Phi-3.5, and Qwen1.5-MoE, respectively, even though the expert hit ratio slightly drops by 2%–3%. This result shows that maximizing prediction accuracy alone does not necessarily maximize inference efficiency. By reducing prediction frequency and amortizing prediction cost across multiple layers, DoMoE lowers runtime overhead enough to offset occasional mispredictions. The gain from prediction optimization is therefore smaller than that from domain-aware routing tables, but it is consistent across all evaluated models. Overall, DoMoE achieves better end-to-end efficiency through a balanced accuracy–overhead trade-off.

4.3 Sensitivity Analysis

We further study the impact of expert routing table size on throughput and expert hit ratio by varying the maximum number of retained entries from 100 to 3000 on Mixtral-8×7B and Phi-3.5. As shown in Figure 5, increasing the routing table size consistently improves expert hit ratio for both models. On Mixtral-8×7B, the hit ratio increases from 43.8% at 100 entries to 69.9% at 3000 entries, corresponding to a 1.6× improvement. Similarly, on Phi-3.5, the hit ratio rises from 53.8% to 73.4%, yielding a 1.36× improvement. These results indicate that larger routing tables preserve more domain-relevant routing signals, leading to more accurate expert activation prediction.

In contrast, throughput exhibits a non-monotonic trend as routing table size increases. For both models, throughput improves as the table size grows up to 2000 entries, where performance peaks. On Mixtral-8×7B, throughput increases from 1.25 tokens/s at 100 entries to 1.43 tokens/s at 2000 entries, achieving a 1.14× speedup. Phi-3.5 shows a similar pattern, with throughput rising from 2.03 to 2.32 tokens/s, corresponding to a 1.14× improvement. However, further increasing the table size to 3000 entries leads to a throughput degradation of approximately 7%–8% for both models, despite improvements in hit ratio. This highlights a trade-off between prediction accuracy and lookup overhead, as excessively large routing tables incur non-negligible similarity matching costs that offset the benefits of higher accuracy.

5 Conclusion

We propose DoMoE, a domain-aware MoE inference system that improves expert offloading through domain-specific routing tables and adaptive prediction. By restricting semantic matching to domain-relevant routes and balancing accuracy against prediction cost, DoMoE avoids the low accuracy of trajectory-based predictors and the high overhead of semantic-based methods. Across diverse MoE models and workloads, DoMoE achieves 1.32× higher throughput and a 1.22× improvement in expert hit ratio, demonstrating its effectiveness for practical MoE deployment [Tang *et al.*, 2024]. Future work will extend DoMoE to concurrent serving under limited GPU memory.

Acknowledgements

The work described in this paper is partially supported by National Natural Science Foundation of China under Grant U24B20149, Taishan Scholars Program under Grant tsqn202408009, and Major Basic Research Program of Shandong Provincial Natural Science Foundation under Grant ZR2025ZD18.

References

[Abdin *et al.*, 2024] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, and Ammar Ahmad. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

[AlicanKiraz0, 2025] AlicanKiraz0. Agentic-chain-of-thought-coding-sft-dataset. https://huggingface.co/datasets/AlicanKiraz0/Agentic-Chain-of-Thought-Coding-SFT-Dataset, 2025. Accessed: 2026-05-25.

[Casanueva *et al.*, 2020] Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*, mar 2020. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

[Fieberg *et al.*, 2024] Christian Fieberg, Lars Hornuf, David Streich, and Maximilian Meiler. Using large language models for financial advice. http://dx.doi.org/10.2139/ssrn.4850039, 2024.

[Hadia *et al.*, 2025] Aliyya Hadia, Tariq Afifa, and Fawzi Gamal. Moe at scale: From modular design to deployment in large-scale machine learning systems. *Preprints*, April 2025.

[He *et al.*, 2023] Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. Merging experts into one: Improving computational efficiency of mixture of experts. *arXiv preprint arXiv:2310.09832*, 2023.

[Hu *et al.*, 2025a] Xing Hu, Zhixuan Chen, Dawei Yang, Zukang Xu, Chen Xu, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. *arXiv preprint arXiv:2505.03804*, 2025.

[Hu *et al.*, 2025b] Yitao Hu, Xiulong Liu, Guotao Yang, Linxuan Li, Kai Zeng, Zhixin Zhao, Sheng Chen, Laiping Zhao, Wenxin Li, and Keqiu Li. Tightllm: Maximizing throughput for llm inference via adaptive offloading policy. *IEEE Transactions on Computers*, 2025.

[Hwang *et al.*, 2024] Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, and Mao Yang. Pre-gated moe: An algorithm-system co-design for fast and scalable mixture-of-expert inference. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pages 1018–1031. IEEE, 2024.

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

[Kim *et al.*, 2023] Young Jin Kim, Raffy Fahim, and Hany Hassan Awadalla. Mixture of quantized experts (moqe): Complementary effect of low-bit quantization and robustness. *arXiv preprint arXiv:2310.02410*, 2023.

[Kong *et al.*, 2025] Rui Kong, Yuanchun Li, Weijun Wang, Linghe Kong, and Yunxin Liu. Serving moe models on resource-constrained edge devices via dynamic expert swapping. *IEEE Transactions on Computers*, 2025.

[Li *et al.*, 2024] Pingzhi Li, Xiaolong Jin, Zhen Tan, Yu Cheng, and Tianlong Chen. Quantmoe-bench: Examining post-training quantization for mixture-of-experts. *arXiv preprint arXiv:2406.08155*, 2024.

[Lightman *et al.*, 2023] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[Liu *et al.*, 2025] Defu Liu, Yixiao Zhu, Zhe Liu, Yi Liu, Changlin Han, Jinkai Tian, Ruihao Li, and Wei Yi. A survey of model compression techniques: past, present, and future. *Frontiers in Robotics and AI*, 2025.

[Lu *et al.*, 2024] Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. *arXiv preprint arXiv:2402.14800*, 2024.

[Ma *et al.*, 2025] Songkai Ma, Zhaorui Zhang, Sheng Di, Benben Liu, Xiaodong Yu, Xiaoyi Lu, and Dan Wang. Compression error sensitivity analysis for different experts in moe model inference. In *Proceedings of the SC'25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 339–348, 2025.

[Maity and Saikia, 2025] Subhankar Maity and Manob Jyoti Saikia. Large language models in healthcare and medical applications: A review. *Bioengineering (Basel, Switzerland)*, 2025.

[Masoudnia and Ebrahimpour, 2014] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 2014.

[Tang *et al.*, 2024] Peng Tang, Jiacheng Liu, Xiaofeng Hou, Yifei Pu, Jing Wang, Pheng-Ann Heng, Chao Li, and Minyi Guo. Hobbit: A mixed precision expert offloading system for fast moe inference. *arXiv preprint arXiv:2411.01433*, 2024.

[Team, 2024] Qwen Team. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters. https://qwenlm.github.io/blog/qwen-moe/, 2024. Accessed: 2026-05-25.

[Team, 2025] Qwen Team. Qwen3 technical report. https://arxiv.org/abs/2505.09388, 2025. Accessed: 2026-05-25.

[Xie *et al.*, 2025] Haojie Xie, Yirong Chen, Xiaofen Xing, Jingkai Lin, and Xiangmin Xu. PsyDT: Using LLMs to construct the digital twin of psychological counselor with personalized counseling style for psychological counseling. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1081–1115. Association for Computational Linguistics, July 2025.

[Xu *et al.*, 2025] Fengyi Xu, Jun Ma, and Nan Li. Large language model applications in disaster management: An interdisciplinary review. *International Journal of Disaster Risk Reduction*, 127:105642, 2025.

[Xu *et al.*, 2026] Yunting Xu, Jiacheng Wang, Ruichen Zhang, Changyuan Zhao, Dusit Niyato, Jiawen Kang, Zehui Xiong, Bo Qian, Haibo Zhou, Shiwen Mao, Abbas Jamalipour, Xuemin Shen, and Dong In Kim. Mixture of experts for decentralized generative ai and reinforcement learning in wireless networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials*, 28:4051–4085, 2026.

[Xue *et al.*, 2024] Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh Marina. Moe-infinity: Offloading-efficient moe model serving. *arXiv preprint arXiv:2401.14361*, 2024.

[Yang *et al.*, 2024] Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. Moe-i^2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. *arXiv preprint arXiv:2411.01016*, 2024.

[Yang *et al.*, 2025] Aiqiang Yang, Jie Liu, Bo Yang, Zeyao Mo, and Keqin Li. Balancing communication overhead and accuracy in compression integration: a survey: A. yang et al. *The Journal of Supercomputing*, 81(8):964, 2025.

[Yu *et al.*, 2025] Hanfei Yu, Xingqi Cui, Hong Zhang, and Hao Wang. fmoe: Fine-grained expert offloading for large mixture-of-experts serving. *arXiv preprint arXiv:2502.05370*, 2025.

[Yue *et al.*, 2024] Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, et al. Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval. In *International Conference on Database Systems for Advanced Applications*, pages 304–321. Springer, 2024.

[Yuksel *et al.*, 2012] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012.

[Zhang *et al.*, 2025] Runhua Zhang, Hongxu Jiang, Wei Wang, and Jinhao Liu. Optimization methods, challenges, and opportunities for edge inference: A comprehensive survey. *Electronics*, 2025.

[Zhong *et al.*, 2024] Shuzhang Zhong, Ling Liang, Yuan Wang, Runsheng Wang, Ru Huang, and Meng Li. Adapmoe: Adaptive sensitivity-based expert gating and management for efficient moe inference. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2024.