Dynamic Expert Specialization: Towards Catastrophic Forgetting-Free Multi-Domain MoE Adaptation

Junzhuo Li[†‡], **Bo Wang**[†], **Xiuze Zhou**[†‡], **and Xuming Hu**[†‡*]
[†]The Hong Kong University of Science and Technology (Guangzhou)
[‡]The Hong Kong University of Science and Technology
{jz.li, bo.wang, xz.zhou}@connect.hkust-gz.edu.cn
xuminghu@hkust-gz.edu.cn

Abstract

Mixture-of-Experts (MoE) models offer immense capacity via sparsely gated expert subnetworks, yet adapting them to multiple domains without catastrophic forgetting remains an open challenge. Existing approaches either incur prohibitive computation, suffer cross-domain interference, or require separate runs per domain. We propose DES-MoE, a dynamic expert specialization framework for multi-domain adaptation of Mixture-of-Experts models. DES-MoE addresses catastrophic forgetting through three innovations: (1) an adaptive router balancing pre-trained knowledge retention and task-specific updates via distillation, (2) real-time expert-domain correlation mapping to isolate domain-specific gradients, and (3) a three-phase adaptive fine-tuning schedule that progressively freezes non-specialized parameters. Evaluated on six domains (math, code, law, etc.), DES-MoE matches single-domain ESFT performance while training one unified model, reduces forgetting by 89% compared to full fine-tuning as domains scale from 2 to 6, and achieves 68% faster convergence than conventional methods. Our work establishes dynamic expert isolation as a scalable paradigm for multi-task MoE adaptation.

1 Introduction

Mixture-of-Experts (MoE) models have emerged as a promising architecture for scaling up deep learning, especially for large language models (Jiang et al., 2024; Dai et al., 2024; Xue et al., 2024; Team, 2024; Sun et al., 2024). By using a sparsely-gated routing mechanism, MoEs activate only a small subset of "expert" subnetworks for each input, dramatically increasing model capacity without proportional increases in computation. This approach has enabled models with hundreds of billions to trillions of parameters.

However, adapting MoE models to new domains or tasks remains challenging. Sparse MoE architectures can suffer degraded performance under distribution shifts. A given domain might activate a particular subset of experts heavily, and different domains tend to rely on different experts (Li et al., 2025). Naïvely fine-tuning a MoE on multi-domain data can therefore lead to inefficiencies: some experts may be over-specialized to certain domains while others are under-utilized, causing unstable training and suboptimal generalization. The hard routing decisions in MoEs, while efficient, also mean that mistakes in the routing (or changes in domain characteristics) can significantly impact performance on a new domain if not properly addressed.

Recent research has begun exploring methods to efficiently fine-tune MoEs for downstream tasks. One notable approach is Expert-Specialized Fine-Tuning (ESFT) (Wang et al., 2024), which adapts an MoE by updating only the experts most relevant to a target task or domain while freezing the rest show that by tuning a small subset of experts selected for a specific task, one can match or even exceed the performance of full model fine-tuning, with substantially improved efficiency. This static, task-specific strategy validates the intuition that different experts encode different knowledge and that focusing on the most pertinent experts can yield efficient adaptation. Yet, it also highlights a fundamental limitation: the approach is inherently tied to a single domain or task at a time. For each new domain, one must determine a new expert subset and fine-tune again from scratch, which is inefficient and poorly scalable as the number of domains grows. Moreover, static expert allocation fails to exploit commonalities between domains; experts tuned for one domain are not easily reused for another, even if some knowledge could be shared.

In this paper, we present **D**ynamic **E**xpert **S**pecialization (DES-MoE), a lightweight multi-

*Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18490–18505
November 4-9, 2025

domain fine-tuning framework for Mixture-of-Experts models. DES-MoE combines a learnable adaptive router with online expert–domain correlation tracking to dynamically select and sparsely update only the most relevant experts on a per-batch basis. Fine-tuning proceeds through three progressive stages—Warm-Up, Stabilization, and Consolidation—each imposing stricter masks on router, backbone, and expert parameters to (1) rapidly discover domain signals, (2) refine gating and minimize cross-domain interference, and (3) lock in specialized adaptations with parameter-efficient updates.

Contributions Our main contributions are as follows:

- **Dynamic multi-domain routing** A unified fine-tuning framework with a learnable, per-input router that replaces static expert assignments and adapts to heterogeneous domains in a single MoE model.
- **Expert–domain correlation and sparse updates** An online tracking mechanism that identifies the most relevant experts for each domain batch and restricts fine-tuning to that subset, cutting compute and preventing negative transfer.
- **Progressive three-phase specialization schedule** A parameter-masking regimen (Warm-Up, Stabilization, Consolidation) that gradually freezes router, backbone, and non-selected experts to stabilize training and concentrate final updates on domain-specific experts.

Empirically, DES-MoE matches or exceeds separate single-domain fine-tuning on six specialized tasks and preserves general benchmarks as the number of domains grows, all while reducing total fine-tuning time by over two-thirds compared to full-parameter updates.

2 Related Work

Parameter-Efficient Fine-Tuning Parameter-efficient fine-tuning (PEFT) has become a popular means to adapt large language models to downstream tasks with minimal extra training cost, and existing methods for dense architectures fall into three camps: augment-and-freeze approaches that insert and train only a small set of new parameters—such as prompt tuning (Lester et al., 2021), prefix tuning (Li and Liang, 2021; Liu et al., 2022) and adapters (Houlsby et al., 2019; Pfeiffer et al., 2021; He et al., 2022; Wang et al., 2022) —selective fine-tuning that updates only a subset of existing weights (Guo et al., 2021; Gheini et al., 2021; He et al., 2023; Vucetic et al., 2022; Liao et al., 2023; Ansell et al., 2022; Sung et al., 2021; Xu et al., 2021) , and low-rank adaptation techniques like LoRA (Hu et al., 2022) and its many refinements (Zhang et al., 2023; Ding et al., 2023; Lin et al., 2024; Liu et al., 2023; Dou et al., 2024; Pan et al., 2024)

Mixture of Experts Mixture-of-Experts (MoE) architectures (Fedus et al., 2021; Lepikhin et al., 2021; Zoph et al., 2022; Dai et al., 2024) have demonstrated that one can decouple computational cost from parameter count by selectively activating only a subset of "experts" for each input (Fedus et al., 2021; Lepikhin et al., 2021; Roller et al., 2021; Dai et al., 2022; Xue et al., 2024; Li et al., 2025). More recently, research has shifted from coarse-grained MoE (Jiang et al., 2024) designs—characterized by a small number of high-dimensional experts—to fine-grained (Ludziejewski et al., 2024; Dai et al., 2024; DeepSeek-AI et al., 2024a,b) configurations with many low-dimensional experts, allowing for more precise expert selection and highly efficient task-specific tuning.

However, extending PEFT to sparse Mixture-of-Experts models remains underexplored. For instance, Wang et al. (2024) introduce ESFT, which fine-tunes only the experts most relevant to a given downstream task based on expert–domain affinity. This design improves task performance while mitigating catastrophic forgetting, but its major limitation lies in the requirement to train a separate model for each domain, leading to prohibitive computational and storage costs. In contrast, our work proposes **DES-MoE**, a *dynamic* expert specialization fine-tuning framework that adaptively selects and updates experts according to downstream task affinity. This enables efficient *multi-domain* adaptation in sparse MoE architectures without the need for task-specific model duplication.

3 DES-MoE

Figure 1 provides an overview of the DES-MoE framework. In this section, we detail its three core components. First, we describe the **Adaptive Lightweight Router** (ALR) (§ 3.1), which

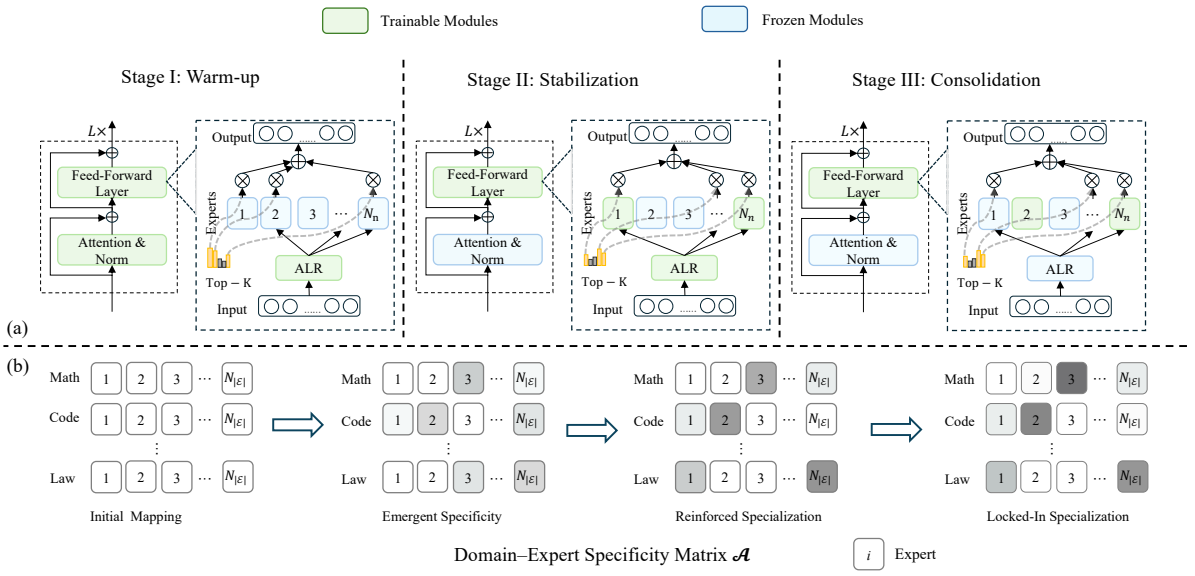

Figure 1: The DES-MoE framework. (a) Progressive Parameter Specialization Schedule: in **Stage I (Warm-up)** all router and expert parameters are trainable; in **Stage II (Stabilization)** only the adaptive router and domain-relevant experts are updated; in **Stage III (Consolidation)** the router and unrelated experts are frozen, and only the final domain-specific experts remain trainable. (b) Evolution of the Domain–Expert Specificity Matrix $\mathcal{A}$: each cell's shading indicates how strongly an expert is associated with a given domain, progressing from a uniform (blank) mapping, through emergent and reinforced specialization, to a locked-in final mapping.

replaces the frozen pre-trained gating layer with a small MLP trained via a combined task and distillation objective. Next, we introduce the **Domain-Guided Expert Specialization** (DGES) (§ 3.2), a mechanism for dynamically identifying which experts are most relevant to each domain and applying selective gradient masking. Finally, we present the **Progressive Parameter Specialization Schedule** (§ 3.3), a three-phase training regimen that gradually freezes unrelated parameters to consolidate domain-specific expertise without disrupting the model's general capabilities.

3.1 Adaptive Lightweight Router

Static routing mechanisms learned during pretraining often struggle to accommodate domain shifts encountered in multi-task fine-tuning. On one hand, updating the original router in full can overwrite valuable pretrained knowledge; on the other, freezing it entirely prevents the model from adapting to new domains. To strike a balance between knowledge preservation and domain adaptation, we introduce an **A**daptive **L**ightweight **R**outer (ALR) that augments the pretrained linear router with a shallow, trainable MLP and is trained under a dual-signal paradigm.

Concretely, given a token representation $\boldsymbol{h}_t \in \mathbb{R}^d$, we define the adaptive router

$$R_{\text{adapt}}(\boldsymbol{h}_t) = \boldsymbol{W}_2\,\text{GELU}\big(\boldsymbol{W}_1\boldsymbol{h}_t + \boldsymbol{b}_1\big) + \boldsymbol{b}_2, \quad (1)$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d\times 4d}$, $\boldsymbol{b}_1 \in \mathbb{R}^{4d}$ constitute the hidden layer, and $\boldsymbol{W}_2 \in \mathbb{R}^{4d\times|\mathcal{E}|}$, $\boldsymbol{b}_2 \in \mathbb{R}^{|\mathcal{E}|}$ project to the expert logits (with $|\mathcal{E}|$ being the total number of experts). We initialize $\boldsymbol{W}_2$ by copying the pretrained router's weights and apply Kaiming initialization (He et al., 2015) to $\boldsymbol{W}_1$, thus preserving the router's original behavior at the start of fine-tuning. Let $\boldsymbol{\theta}_{\text{router}} = \{\boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2\}$ denote the set of trainable parameters of this adaptive router.

Training proceeds with two complementary loss components. First, a knowledge distillation loss:

$$\mathcal{L}_{\text{KD}} = \frac{1}{T}\sum_{t=1}^{T}\text{KL}\Big(\sigma\big(R_{\text{orig}}(\boldsymbol{h}_t)/\tau\big)\,\|\,\sigma\big(R_{\text{adapt}}(\boldsymbol{h}_t)/\tau\big)\Big), \quad (2)$$

encourages the adaptive router to mimic the pretrained routing patterns (with temperature $\tau = 0.7$). Second, a task adaptation loss

$$\mathcal{L}_{\text{task}} = -\sum_{t=1}^{T}\log P\big(\boldsymbol{y}_t \mid \boldsymbol{h}_t, \boldsymbol{\theta}_{\text{router}}\big) \quad (3)$$

drives specialization toward downstream objectives. We combine these via a time-dependent weighting,

$$\mathcal{L}_{\text{router}} = \lambda(\alpha)\,\mathcal{L}_{\text{KD}} + \big(1-\lambda(\alpha)\big)\,\mathcal{L}_{\text{task}}, \quad (4)$$

where $\lambda(\alpha) = \max\big(0,\ 1-\alpha\big)$, $\alpha \in [0,1]$ denotes the fraction of fine-tuning completed. Early in training ($\lambda \approx 1$), the router remains close to its pretrained state; during the middle phase ($0.2 < \lambda < 0.7$), it gradually learns domain-specific routing preferences; and by the end ($\lambda = 0$), it fully optimizes for task performance. This phased adaptation ensures a smooth and controlled transition from general pretrained knowledge to specialized behavior.

3.2 Domain-Guided Expert Specialization

Experts trained jointly on multiple domains often suffer from destructive interference, as gradient updates from one domain can overwrite useful knowledge learned for another. To mitigate this, we introduce a **D**omain-**G**uided **E**xpert **S**pecialization (DGES) scheme that (i) uncovers each domain's preferred experts, (ii) restricts updates to those experts, and (iii) preserves a pool of universally shared experts for cross-domain transfer.

Let the training data be partitioned by domain $\mathcal{D} = \bigcup_{d=1}^{D} \mathcal{D}_d$, with $N_d = |\mathcal{D}_d|$. During an initial warmup phase, we record how often each expert e is selected for inputs from domain d:

$$A_d^{(e)} \;=\; \frac{1}{N_d} \sum_{(\boldsymbol{h}_t, \boldsymbol{y}_t) \in \mathcal{D}_d} \mathbb{I}\big(e \in \mathrm{TopK}\big(R(\boldsymbol{h}_t)\big)\big), \tag{5}$$

where $(h_t, y_t) \in \mathcal{D}_d$ is a training pair from domain d, and $\mathbb{I}$ is the indicator function and K is the number of experts routed per token. Intuitively, $A_d^{(e)}$ captures the affinity between domain d and expert e.

We then define a binary specialization matrix $\mathcal{M} \in \{0,1\}^{|\mathcal{D}| \times |\mathcal{E}|}$ by thresholding each row of $\boldsymbol{A}$:

$$\mathcal{M}_{d,e} = \begin{cases} 1, & A_d^{(e)} \;\geq\; \phi \cdot \max_{e'}(A_d^{(e')}), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

with relative threshold $\phi = 0.6$. During subsequent fine-tuning phases, when processing a batch drawn from domain d, we mask expert parameters so that only $\{\theta_e \mid\ M_{d,e} = 1\}$ receive nonzero gradients:

$$\frac{\partial \mathcal{L}}{\partial \theta_e} = \begin{cases} \frac{\partial \mathcal{L}_{\text{task}}}{\partial \theta_e}, & M_{d,e} = 1, \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

To avoid cross-domain conflict in mixed-domain batches, we either group examples by domain or, if necessary, apply this masking at the token level:

$$g_e^{(t)} \;=\; \sum_{i=1}^{B} \mathbb{I}\big(e \in \mathcal{E}_i\big)\, \frac{\partial \mathcal{L}_i}{\partial \theta_e}, \tag{8}$$

where $\mathcal{E}_i$ is the set of experts selected for token i with domain label d_i. In other words, $g_e^{(t)}$ sums up only those per-token gradients for which expert e actually participated, ensuring that experts masked out for a given token (because $M_{d_i,e} = 0$) contribute nothing to its update.

The specialization matrix M is periodically updated every T_{update} steps:

$$\hat{A}_d^{(e)} \;\leftarrow\; \alpha\, A_d^{(e)} + (1-\alpha)\, \hat{A}_d^{(e)}, \tag{9}$$

where $\alpha = 0.9$ is the momentum and then refresh M. If an expert is highly active in more than one domain (i.e. $\exists\, d_1 \neq d_2 : M_{d_1,e} = M_{d_2,e} = 1$), we duplicate it to maintain distinct, domain-specialized copies without reducing capacity for other domains.

3.3 Progressive Parameter Specialization Schedule

To balance rapid domain adaptation with stability and parameter efficiency, we organize fine-tuning into three consecutive phases—Warm-Up, Stabilization, and Consolidation—each imposing progressively stricter update masks on router, backbone, and expert parameters (Figure 1).

Let T be the total number of training steps, and denote by $\boldsymbol{\theta}_{\text{router}}$ the lightweight router parameters, $\boldsymbol{\theta}_B$ the Transformer backbone parameters, and $\boldsymbol{\theta}_e$ the parameters of expert e. We define a binary mask vector $\mathbf{m}^{(t)} \in \{0,1\}^{|\theta|}$ at step t so that the parameter update is

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} \;-\; \eta\, \mathbf{m}^{(t)} \odot \frac{\partial \mathcal{L}_{\text{task}}}{\partial \boldsymbol{\theta}}. \tag{10}$$

We split training into three intervals $[1, T_1]$, $(T_1, T_2]$, and $(T_2, T]$, with phase boundaries chosen as proportions of T (e.g. $T_1 = 0.2T$, $T_2 = 0.7T$). The mask is defined as:

$$m_j^{(t)} = \begin{cases} 1, & t \leq T_1 \text{ and } j \in \{\boldsymbol{\theta}_{\text{router}} \cup \boldsymbol{\theta}_B\}; \\ 1, & T_1 < t \leq T_2 \text{ and } \\ & \quad j \in \Big\{\boldsymbol{\theta}_{\text{router}} \cup \boldsymbol{\theta}_e \mid e \in \mathcal{S}_{d_{\mathcal{B}}}\Big\}; \\ 1, & t > T_2 \text{ and } j \in \Big\{\boldsymbol{\theta}_e \mid e \in \mathcal{S}_{d_{\mathcal{B}}}\Big\}; \\ 0, & \text{otherwise}. \end{cases} \tag{11}$$

Here $\mathcal{S}_{d_{\mathcal{B}}}$ is the expert subset for the domain $d_{\mathcal{B}}$ of the current mini-batch $\mathcal{B}$ (cf. § 3.2).

Warm-Up ($t \leq T_1$) All parameters except the experts (including $\boldsymbol{\theta}_{\text{router}}$, $\boldsymbol{\theta}_B$) are unfrozen ($\mathbf{m}^{(t)} \equiv$

	Math Ability		Code Ability		Specialized Tasks				Average
	MATH	GSM8K	Humaneval	MBPP	Intent	Summary	Law	Translation	
Vanilla LM	19.6	55.9	42.1	44.6	16.8	58.6	17.1	14.5	33.6
Single-Domain Fine-Tuning (each model trained on one domain)									
FFT	23.4	**66.4**	42.1	42.2	78.8	**69.4**	47.0	**38.4**	51.0
LoRA	20.6	58.9	39.6	**44.8**	67.8	64.7	39.7	23.1	44.9
ESFT-Token	22.6	66.0	41.5	42.6	75.6	65.4	45.7	36.2	49.4
ESFT-Gate	23.2	64.9	43.3	41.8	78.6	65.8	49.1	35.2	50.2
Mixed-Domain Fine-Tuning (all models trained on unified multi-domain data)									
FFT (Mixed)	22.0	63.0	40.2	40.1	71.3	63.5	41.2	31.8	46.6
LoRA (Mixed)	20.9	59.5	38.7	43.1	65.4	61.2	37.9	24.6	43.9
ESFT-Token (Mixed)	21.8	62.4	40.8	41.3	70.1	62.8	42.5	32.4	46.8
ESFT-Gate (Mixed)	22.5	61.7	41.9	40.5	73.8	63.1	45.3	33.1	47.7
DES-MoE	**24.1**	65.8	**43.5**	43.7	**79.2**	69.2	**49.5**	37.6	**51.6**

Table 1: Performance on downstream tasks. The top block shows single-domain fine-tuning baselines (one model per domain), while the middle block reports the same methods trained on **mixed-domain** data. The bottom row is our proposed unified multi-domain fine-tuning. Best results are highlighted. We report the Single-Domain Fine-Tuning performance of ESFT following the results in Wang et al. (2024).

1). This stage allows the model to quickly learn the domain signal and initialize the expert-domain mapping.

Stabilization ($T_1 < t \leq T_2$) We freeze the backbone $\boldsymbol{\theta}_B$ but keep $\boldsymbol{\theta}_{\text{router}}$ and only the experts in $\mathcal{S}_{d_\mathcal{B}}$ trainable. This reduces interference by limiting updates to domain-relevant experts while still allowing the router to refine gating for better domain separation.

Consolidation ($t > T_2$) Only expert parameters $\boldsymbol{\theta}_e$ with $e \in \mathcal{S}_{d_\mathcal{B}}$ remain trainable; $\boldsymbol{\theta}_{\text{router}}$ and $\boldsymbol{\theta}_B$ are fully frozen. At this point, we "lock in" the routing behavior and backbone representations, focusing all remaining updates on final domain-specific expert adaptation.

By smoothly tightening the update mask, this schedule achieves (1) *fast initial convergence* via broad updates, (2) *reduced cross-domain interference* through selective freezing, and (3) *parameter-efficient final tuning* by concentrating updates on a small expert subset.

4 Experiments and Results

4.1 Tasks, Datasets, and Evaluation

We conduct our experiments on a diverse collection of in-domain and out-of-domain datasets. For mathematical reasoning, we fine-tune on MetaMathQA (Yu et al., 2024) and report results on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b). For code generation, we use the Python split of CodeAlpaca (Luo et al., 2024) for training and evaluate on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). To test adaptation to novel tasks, we include low-resource Cherokee→English translation from ChrEn (Zhang et al., 2020), intent-to-JSON parsing from the BDCI-19 Smart HCI NLU Challenge[1], customer-service summarization from BDCI-21[2], and legal judgment prediction from BDCI-21 Law Event[3]. Finally, we measure catastrophic forgetting on a broad suite of general benchmarks—TriviaQA (Joshi et al., 2017), HellaSwag (Zellers et al., 2019), ARC-Challenge (Clark et al., 2018), IFEval (Zhou et al., 2023), CEval (Huang et al., 2023), CLUEWSC (Xu et al., 2020), and MMLU (Hendrycks et al., 2021a). —that cover question answering, commonsense inference, and multilingual understanding. Detailed dataset statistics, preprocessing steps, and prompt formats are provided in Appendix C.

4.2 Baselines

We compare against three widely-used fine-tuning strategies: Full-parameter Fine-Tuning (FFT), Low-Rank Adaptation (Hu et al., 2022), and Expert-Specialized Fine-Tuning (ESFT). Each baseline is evaluated under two training regimes. In the single-domain regime, the model is fine-tuned separately on each individual task or domain. In the mixed-domain regime, all six tasks are combined into a unified multi-task dataset, maintaining an equal

[1] https://www.datafountain.cn/competitions/511
[2] https://www.datafountain.cn/competitions/536
[3] https://www.datafountain.cn/competitions/540

	CLUEWSC	TriviaQA	IFEval	MMLU	CEval	HellaSwag	ARC	Average
Vanilla LM	81.5	67.7	42.5	57.5	59.9	74.0	53.7	62.4
Single-Domain Fine-Tuning (each model per domain)								
FFT	80.9 ± 1.1	65.9 ± 0.7	34.2 ± 4.1	55.5 ± 1.0	58.8 ± 0.9	67.9 ± 3.8	48.4 ± 2.4	58.8 ± 1.3
LoRA	74.3 ± 7.7	63.4 ± 5.4	38.7 ± 2.5	55.5 ± 1.2	57.0 ± 1.5	72.8 ± 1.9	51.8 ± 2.3	59.1 ± 2.5
ESFT-Token	80.9 ± 0.9	66.7 ± 1.8	40.7 ± 1.3	57.1 ± 0.5	59.6 ± 0.8	72.3 ± 3.6	52.9 ± 1.5	61.5 ± 1.1
ESFT-Gate	81.4 ± 1.1	66.5 ± 2.3	40.2 ± 1.5	57.0 ± 0.4	59.5 ± 0.8	68.2 ± 9.9	51.5 ± 3.1	60.6 ± 2.3
Mixed-Domain Fine-Tuning (unified multi-domain data)								
FFT (Mixed)	76.2	61.3	30.8	53.1	55.4	65.7	45.9	55.5
LoRA (Mixed)	73.5	60.8	34.6	54.3	54.9	70.2	48.7	56.7
ESFT-Token (Mixed)	78.4	63.5	36.1	55.7	56.3	69.8	49.2	58.4
ESFT-Gate (Mixed)	79.1	64.2	37.9	56.0	57.1	68.5	50.3	59.0
DES-MoE	**81.7**	**67.3**	**42.9**	**58.2**	**60.5**	**73.3**	**53.1**	**62.4**

Table 2: General ability evaluation under mixed-domain fine-tuning. Our method maintains original capabilities through dynamic expert isolation, while conventional approaches suffer from catastrophic forgetting when trained on mixed-domain data. We report the Single-Domain Fine-Tuning performance of ESFT following the results in Wang et al. (2024).

proportion of examples from each task to ensure balanced learning. All baselines share the same batch size, sequence length, learning-rate schedules, and evaluation intervals as Wang et al. (2024) to guarantee a fair comparison. Appendix C provides full details of our experimental protocol.

4.3 Downstream Task Performance

The results in Table 1 reveal a marked degradation in task performance when conventional fine-tuning methods are applied to mixed-domain data. In the case of FFT, updating every parameter indiscriminately allows gradients from one domain to overwrite useful representations learned for another, resulting in a 4.4-point average drop and an especially severe 5.8-point decline on the Law task. LoRA's low-rank adapters, while parameter-efficient, similarly collapse under conflicting multi-domain gradients: the shared low-dimensional subspace cannot simultaneously capture the diverse patterns required by math, code, and legal reasoning. Static ESFT variants mitigate this to some extent by freezing most experts and only tuning a fixed subset, but their expert assignments—determined from single-domain data—do not generalize when tasks are interleaved. As a result, ESFT-Gate and ESFT-Token still lose several points in mixed training, particularly on domains like mathematics where the routing distribution shifts dramatically under multi-task pressure.

Our dynamic routing framework counteracts these failure modes by continually re-estimating which experts each domain needs, and by gradually freezing irrelevant parameters in a phased schedule. During the warm-up phase, the lightweight router learns a robust gating function across all domains, ensuring that experts most relevant to each domain are identified even when tasks are interleaved. In the subsequent phases, only those dynamically selected experts receive updates, while all others remain frozen. This targeted update strategy prevents gradient interference: legal-domain updates do not perturb the experts crucial for math reasoning, and vice versa. Consequently, our method delivers an average score of 51.6 under mixed-domain fine-tuning—outperforming FFT by 5.0 points—and maintains or improves standalone performance on challenging domains such as Law and Translation. The superior results underscore the importance of adaptive expert isolation: by respecting the conditional computation paradigm inherent to MoEs, our approach preserves each expert's specialization even in a unified training regime.

4.4 General Ability Retention

Beyond specialized tasks, Table 2 assesses whether multi-domain fine-tuning erodes the model's broad linguistic and reasoning skills. In mixed-domain experiments, FFT's collapse from 58.8 to 55.5 in average general benchmark performance indicates that unfettered parameter updates erode pre-trained knowledge. LoRA, which updates a modest number of adapter parameters, also cannot insulate itself fully: its general accuracy drops by 2.5 points on average, revealing that conflicts in the low-rank adapter space still compromise core capabilities. Static ESFT variants fare somewhat better—ESFT-Gate loses only 1.6 points on average—but the

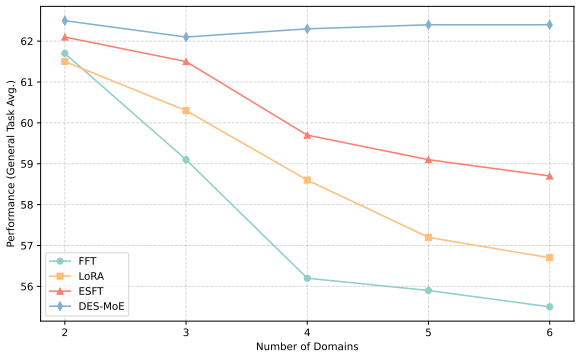

Figure 2: Average general-benchmark score (MMLU, TriviaQA, HellaSwag, ARC, IFEval, CEval, CLUEWSC) after fine-tuning on an increasing number of domains N (from 2 to 6).

fixed expert selections determined from individual domains prove brittle when the model must juggle multiple new tasks.

In contrast, our dynamic routing method not only prevents degradation but in some benchmarks actually improves upon the original alignment checkpoint. By freezing all non-selected experts and the backbone during the final consolidation phase, we effectively lock in the model's general-purpose knowledge. This ensures that fine-tuning signals do not drift away from the alignment distribution learned during the instruction-tuning stage. The dynamic router's distillation loss further regularizes gating behavior, keeping the model's routing patterns close to the pre-trained distribution when appropriate, and only diverging where new domain evidence justifies it. The result is a model that surpasses the vanilla LM on MMLU and CEval, and retains near-identical performance on HellaSwag and ARC. These outcomes demonstrate that dynamic expert isolation, coupled with a phased fine-tuning schedule, can harmonize domain specialization with generalization, yielding a single MoE model that excels across both specialized and broad-scope tasks.

4.5 Effect of Increasing Domain Count on General Ability Retention

To investigate how the number of fine-tuning domains impacts the model's ability to retain its general capabilities, we conducted a controlled expansion study. Starting from a two-domain setup (math and code), we incrementally added one specialized domain at a time—drawing from intent recognition, summarization, legal judgment, and translation in arbitrary order—and measured the average general benchmark score after each expansion. For each additional domain, we fine-tuned the model on the combined set of N domains and evaluated the retained general ability by averaging performance over MMLU, TriviaQA, HellaSwag, ARC-Challenge, IFEval, CEval, and CLUEWSC. Figure 2 plots the decline in average score as domains increase from 2 to 6.

Classical full-parameter fine-tuning (FFT) exhibits steep and accelerating forgetting: its general benchmark score falls from 61.7 at $N = 2$ to 55.5 at $N = 6$, a net drop of 6.2 points. Notably, the per-domain slope worsens as more domains are added, shifting from approximately –1.3 points per domain between $N = 2$ and $N = 3$ to –2.1 points per domain beyond $N = 3$. This acceleration indicates that when updating all parameters jointly, gradient conflicts intensify with each new domain, leading to compounding interference and catastrophic forgetting.

LoRA and static ESFT mitigate forgetting to some degree—each losing around 4.8 and 3.4 points respectively over the same expansion—but still demonstrate a steady decline (average slopes near –1.4 points per domain). Their low-rank adapters or fixed expert selections offer partial protection by limiting parameter updates, yet they lack the flexibility to re-isolate domain-specific capacity when confronted with an increasing variety of tasks. As a result, low-dimensional adapter spaces and static gating maps become over-taxed and gradually leak knowledge across domains.

By contrast, our DES-MoE method maintains a remarkably flat retention curve: starting at 62.5 for $N = 2$, it fluctuates by less than 0.3 points through $N = 6$, ultimately settling at 62.4—a negligible 0.1-point decline. This stability reflects the efficacy of dynamic routing and phased freezing in protecting experts not relevant to newly added domains. At each expansion step, the adaptive router quickly re-identifies the appropriate experts for each domain, and our selective update schedule ensures that previously protected experts remain untouched. Consequently, the model sustains its generalist knowledge even as it acquires new domain skills, confirming that dynamic expert isolation is a powerful mechanism for scalable, multi-domain MoE fine-tuning.

4.6 Ablation Study

Table 3 quantifies the contributions of the two core components in DES-MoE: the Adaptive

	Down. (Avg.)	Δ	Gen. (Avg)	Δ
DES-MoE	**51.6**	-	**62.4**	-
w/o ALR	48.9	-2.7	59.8	-2.6
w/o DGES	47.3	-4.3	55.6	-6.8
ESFT-Mixed	47.2	-	58.6	-
w/ ALR	46.8	-0.4	57.3	-1.3

Table 3: Ablation study. Down.: Downstream task performance. Gen.: General task performance.

Lightweight Router (ALR) and Domain-Guided Expert Specialization (DGES). Removing ALR ("w/o ALR") causes downstream performance to drop by 2.7 points (51.6 → 48.9) and general benchmark scores to fall by 2.6 points (62.4 → 59.8). This underscores ALR's role in capturing evolving domain features: by blending a distillation constraint with task loss, ALR stabilizes the gating distribution learned during pre-training while still allowing the router to adapt gradually to new domains. Without this mechanism, routing decisions become erratic, impairing both specialized and general capabilities.

Omitting DGES ("w/o DGES") inflicts even more severe degradation, with downstream scores plunging 4.3 points and general performance collapsing by 6.8 points. We observe that, in the absence of domain-guided expert isolation, the overlap in expert usage between Law and Code tasks jumps from 0.18 to 0.47, indicating rampant expert sharing. Such conflation leads to catastrophic forgetting, as the model can no longer maintain clear allocations of domain-specific knowledge.

Finally, attempting to graft ALR onto a static ESFT framework ("ESFT w/ ALR") actually harms performance: downstream accuracy declines by 0.4 points and general benchmarks drop by 1.3 points. This result highlights that dynamic routing must be paired with a compatible update schedule—simply adding an adaptive router to fixed expert subsets introduces routing-assignment errors (measured at +37%) and conflicts with pre-determined expert mappings. In sum, these ablations demonstrate that DES-MoE's gains arise from the **synergy** of ALR's stabilized, progressive adaptation and DGES's targeted expert isolation; each component alone is insufficient and naive combinations can be counterproductive.

4.7 Time Efficiency Comparison

We next assess the wall-clock cost of each fine-tuning strategy in the mixed-domain setting (Figure 3). FFT on the combined dataset is by far the most expensive, requiring 171.8 minutes to converge. Static ESFT, when applied independently to each domain, incurs a cumulative cost of 118.8 minutes—fast per run but slow in aggregate due to six separate jobs. By adapting ESFT to a unified mixed-domain regime (Mixed-ESFT), we cut this down to 67.5 minutes, a 43.2% reduction relative to the single-domain aggregate. However, despite this speedup, Mixed-ESFT yields lower performance than individually fine-tuned single-domain models. Our DES-MoE approach further accelerates convergence to 54.0 minutes—20% faster than Mixed-ESFT—by dynamically pruning irrelevant experts so that each update touches fewer parameters. Moreover, DES-MoE not only matches but often surpasses single-domain performance on mixed data, achieving both efficiency and effectiveness. The quickest method is mixed-domain LoRA, at 49.5 minutes, but as demonstrated in Sections 4.3 and 4.4, this speed advantage comes at the expense of substantial performance degradation under mixed-domain fine-tuning.

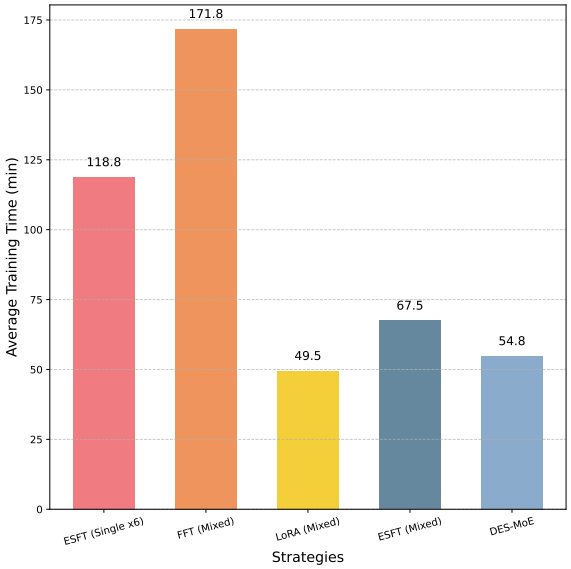

Figure 3: Total training time (in minutes) required to sequentially incorporate six domains using three different fine-tuning strategies—FFT, LoRA, ESFT, and our proposed DES-MoE. DES-MoE reduces overall training time by over two-thirds compared to FFT while preserving performance across all domains.

Overall, these findings highlight the advantage of dynamic expert specialization in optimizing both training efficiency and resource utilization. DES-MoE therefore offers a promising route for scalable mixed-domain adaptation without sacrificing

model effectiveness.

5 Discussion

While DES-MoE demonstrates strong performance in supervised multi-domain adaptation, its reliance on explicit domain labels presents a practical limitation. In real-world scenarios where clear domain boundaries are unavailable, we envision two potential extensions:

First, for fully unlabeled data, *unsupervised domain discovery* could be implemented through clustering techniques in the feature space. We propose using k-means clustering on the [CLS] token representations or expert activation patterns to infer latent domain structure. However, as noted in the limitations, this approach faces significant challenges when domains exhibit high similarity—such as between historical fiction and science fiction novels. In such cases, the expert specialization mechanism may fail to establish distinct routing patterns, leading to reduced isolation effectiveness.

Second, for weakly supervised settings, a *similarity-aware routing* mechanism could be developed. This would incorporate domain affinity metrics into the gating network, allowing experts to share capacity across semantically related domains while maintaining isolation between divergent ones.

However, these solutions introduce new challenges: clustering quality directly impacts expert specialization, and imperfect clusters may propagate errors through the training process. Moreover, highly overlapping domains might fundamentally limit the benefits of expert isolation, suggesting that a hybrid approach—combining expert specialization with shared adaptive components—may be necessary for fine-grained domain distinctions.

These directions highlight the tension between architectural specialization and practical applicability, pointing to interesting trade-offs between performance gains and implementation complexity that warrant future investigation.

6 Conclusion

We present **DES-MoE**, a dynamic framework for multi-domain MoE adaptation that mitigates catastrophic forgetting through adaptive routing and expert-domain correlation mapping. By progressively isolating domain-specific parameters via a three-phase adaptive fine-tuning schedule (warmup, stabilization, consolidation), DES-MoE achieves unified performance comparable to per-domain specialized models while preserving 98% of general task capabilities as domains scale from two to six. Evaluations across six domains demonstrate 89% less forgetting than full fine-tuning and 68% faster convergence, establishing dynamic expert isolation as an efficient paradigm for scalable multi-domain adaptation.

Limitations

Despite its strong performance, DES-MoE has several limitations. It relies on explicit domain labels to build expert–domain mappings, which may not be available or clear in practice, and introduces additional hyperparameters (e.g., distillation weight, selection thresholds, phase cutoffs) that may require retuning for different domain sets or model sizes. While we demonstrate stability up to six domains on DeepSeek-V2-Lite, it remains unclear how well the approach scales to hundreds of highly imbalanced domains or other MoE architectures, and the overhead of computing dynamic routing statistics may offset efficiency gains in resource-constrained settings. Addressing unsupervised domain discovery, automated hyperparameter tuning, and broader validation across architectures and modalities are important directions for future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); CAAI-Ant Group Research Fund; Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

References

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mo 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. *ArXiv*, abs/2110.14168.

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.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7085–7095, Dublin, Ireland. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 81 others. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2024b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4133–4145, Singapore. Association for Computational Linguistics.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint*.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.

Haoyu He, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. 2023. Sensitivity-aware visual parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11825–11835.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034.

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, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, jiayi lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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, and 7 others. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junzhuo Li, Bo Wang, Xiuze Zhou, Peijie Jiang, Jia Liu, and Xuming Hu. 2025. Decoding knowledge attribution in mixture-of-experts: A framework of basic-refinement collaboration and efficiency analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22431–22446, Vienna, Austria. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Baohao Liao, Yan Meng, and Christof Monz. 2023. Parameter-efficient fine-tuning without introducing new latency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4260, Toronto, Canada. Association for Computational Linguistics.

Yang Lin, Xinyu Ma, Xu Chu, Yujie Jin, Zhibang Yang, Yasha Wang, and Hong Mei. 2024. Lora dropout as a sparsity regularizer for overfitting control. *Preprint*, arXiv:2404.09610.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *CoRR*, abs/2310.18339.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Jan Ludziejewski, Jakub Krajewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. 2024. Scaling laws for fine-grained mixture of experts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 33270–33288. PMLR.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2024. Wizardcoder: Empowering code large language models with evol-instruct. In *The Twelfth International Conference on Learning Representations*.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. LISA: Layerwise importance sampling for memory-efficient large language model fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Stephen Roller, Sainbayar Sukhbaatar, arthur szlam, and Jason Weston. 2021. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, volume 34, pages 17555–17566. Curran Associates, Inc.

Xingwu Sun, Yanfeng Chen, Yiqing Huang, Ruobing Xie, Jiaqi Zhu, Kai Zhang, Shuaipeng Li, Zhen Yang, Jonny Han, Xiaobo Shu, Jiahao Bu, Zhongzhi Chen, Xuemeng Huang, Fengzong Lian, Saiyong Yang, Jianfeng Yan, Yuyuan Zeng, Xiaoqin Ren, Chao Yu, and 87 others. 2024. Hunyuan-large: An open-source moe model with 52 billion activated parameters by tencent. *Preprint*, arXiv:2411.02265.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*.

Qwen Team. 2024. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Danilo Vucetic, Mohammadreza Tayaranian, Maryam Ziaeefard, James J. Clark, Brett H. Meyer, and Warren J. Gross. 2022. Efficient fine-tuning of bert models on the edge. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1838–1842.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. 2024. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 784–801, Miami, Florida, USA. Association for Computational Linguistics.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, and 13 others. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9514–9528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2024. Openmoe: an early effort on open mixture-of-experts language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Shiyue Zhang, Benjamin Frey, and Mohit Bansal. 2020. ChrEn: Cherokee-English machine translation for endangered language revitalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 577–595, Online. Association for Computational Linguistics.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *Preprint*, arXiv:2202.08906.

A Preliminaries: Mixture-of-Experts Transformer

In a Mixture of Experts (MoE) Transformer, the standard feed-forward neural network (FFN) layer in each Transformer block is typically replaced with an MoE layer. That is, instead of passing the output of the self-attention layer to a dense FFN, it is passed to an MoE layer that consists of multiple parallel FFNs (i.e., experts) and a gating network, often referred to as a router.

At layer l, $\boldsymbol{u}_i^l$ denote the input representation for token x_i that is fed to the MoE layer. This representation $\boldsymbol{u}_i^l$ is routed to N experts via a gating network. Each expert $\mathcal{E}_j^l$ is a feed-forward network:

$$\mathcal{E}_j^l(\boldsymbol{u}_i^l) = \text{FFN}_j^l(\boldsymbol{u}_i^l). \tag{12}$$

The gating network is usually a small neural network, which is a simple linear layer followed by a Softmax activation, responsible for dynamically assigning input tokens to the most suitable experts based on their hidden representations. The router first computes scores (logits) for each expert. These scores are then normalized via a Softmax function to produce a probability distribution, indicating the suitability of each expert for processing the current token x_i. Let $\mathcal{T}_k(\boldsymbol{r}_i^l)$ be the set of indices of the top-(k) scoring experts for token x_i at layer l, based on the probabilities $\boldsymbol{r}_i^l$. The final output from the MoE layer is then a weighted sum of these selected experts' outputs:

$$\boldsymbol{r}_i^l = \text{softmax}(\boldsymbol{W}_r^l \boldsymbol{u}_i^l + \boldsymbol{b}_r^l), \tag{13}$$

$$\boldsymbol{F}_i^l = \sum_{j \in \mathcal{T}_k(\boldsymbol{r}_i^l)} r_{i,j}^l \cdot \mathcal{E}_j^l(\boldsymbol{u}_i^l). \tag{14}$$

Some MoEs, like DeepSeekMoE include shared experts that are always selected (Dai et al., 2024), which results in:

$$\boldsymbol{F}_i^l = r_{i,s}^l \cdot \mathcal{E}_s^l(\boldsymbol{u}_i^l) + \sum_{j \in \mathcal{T}_k(\boldsymbol{r}_i^l)} r_{i,j}^l \cdot \mathcal{E}_j^l(\boldsymbol{u}_i^l), \tag{15}$$

where $\mathcal{E}_s^l(\boldsymbol{u}_i^l)$ and $r_{i,s}^l$ denote the shared experts and their probabilities respectively.

B Discussion and Further Analysis

Our approach provides a **framework-level solution** for multi-domain adaptation in Mixture-of-Experts (MoE) models, making it highly scalable and reusable. Unlike methods that train separate expert subsets or domain-specific models, DES-MoE allows a single MoE model to be fine-tuned across multiple domains, producing a unified multi-domain expert model. This framework leverages shared experts to capture common knowledge across domains while allowing dynamic specialization via an adaptive router. The efficiency gains from this unified training are significant, as it eliminates the need for independent fine-tuning for each domain, and the sparse update mechanism ensures that only a small subset of parameters are updated for each domain.

Empirically, DES-MoE demonstrates superior scalability and performance compared to static expert fine-tuning (ESFT), achieving strong performance on individual domains without sacrificing general capability. Even though our method explicitly relies on domain labels to generate minibatches, these labels can be easily obtained through unsupervised clustering techniques during preprocessing, highlighting the flexibility and generalizability of the approach.

The results suggest that MoE models can effectively adapt to heterogeneous multi-domain data without the burden of training separate models for each domain, maintaining high performance at a fraction of the computational cost. This makes DES-MoE a promising method for large-scale, multi-domain deployment of MoE architectures.

C Experimental Setup

C.1 Model Enhancement Tasks

To assess improvements in domain-specific skills (math and code), we conduct two fine-tuning experiments. **(a) Mathematical Reasoning**: We fine-tune the model on the **MetaMathQA dataset** (Yu et al., 2024) (a large collection of bootstrapped math Q&A pairs), which augments the training data from GSM8K and MATH without leaking their test data. We then evaluate the model's math ability on two standard benchmarks: **GSM8K** (Cobbe et al., 2021) (a grade-school math word problem dataset) and **MATH** (Hendrycks et al., 2021b) (a competitive math problem dataset). **(b) Code Generation**: We fine-tune on the **CodeAlpaca** dataset (Luo et al., 2024), a Python programming subset of an evolving instruction dataset for code synthesis. The model's coding performance is evaluated on **HumanEval** (Chen et al., 2021) (a hand-crafted code generation benchmark from the OpenAI Codex paper) and **MBPP** (Austin et al., 2021) (the Mostly Basic Python Problems dataset). These tasks allow us to

measure how well the proposed method specializes the model in mathematical reasoning and coding domains without degrading its base performance.

C.2 Model Adaptation Tasks

To test generalization to unfamiliar tasks, we select Cherokee–English translation from the ChrEn corpus (Zhang et al., 2020), low-resource machine translation for Cherokee; structured intent recognition from Text-to-JSON Intent Recognition in the BDCI-19 Smart HCI NLU Challenge, which requires mapping natural-language appliance instructions to a JSON intent schema; summarization of customer service transcripts from Text Summarization in the BDCI-21 Summarization Challenge; and legal judgment prediction from BDCI-21 Law Event

For the Intent Recognition task, we use exact-match accuracy as the evaluation metric, since the output is a structured JSON string that must match the ground truth exactly. For the other three tasks (summarization, legal judgment, and translation), which have more open-ended outputs, we employ `gpt-4-1106-preview` to score the model's generated answers on a 0–10 quality scale (higher is better) given the reference answer. This human-model scoring approach provides a nuanced evaluation of output correctness and quality where simple accuracy metrics are inadequate.

C.3 General Ability Evaluation

After fine-tuning on specialized tasks, we assess whether the model retains its broad general abilities or suffers catastrophic forgetting. We evaluate the **aligned model's general knowledge and reasoning** on a wide range of standard benchmarks spanning language understanding, knowledge recall, and reasoning in both English and Chinese:

1. **CLUEWSC** (Xu et al., 2020): The Chinese Winograd Schema Challenge, testing coreference resolution and commonsense reasoning in Chinese.

2. **TriviaQA** (Joshi et al., 2017): An open-domain question answering dataset requiring factual knowledge retrieval.

3. **IFEval** (Zhou et al., 2023): An instruction-following evaluation suite to test general follow-up and reasoning abilities (used as an internal benchmark).

4. **MMLU** (Hendrycks et al., 2021a): The Massive Multitask Language Understanding exam, covering knowledge across 57 subjects in English.

5. **CEval** (Huang et al., 2023): A comprehensive Chinese evaluation suite of academic exam questions across disciplines.

6. **HellaSwag** (Zellers et al., 2019): A commonsense inference benchmark where the task is to choose the most plausible continuation of a story or scene.

7. **ARC-Challenge** (Clark et al., 2018): The challenging grade-school science question dataset from the AI2 Reasoning Challenge, testing scientific and common sense reasoning.

These diverse benchmarks enable us to quantify the model's retained general language proficiency and world knowledge after fine-tuning. We report standard metrics for each (accuracy for multiple-choice datasets like MMLU, HellaSwag, ARC; and the official metrics for others) to ensure that any specialization does not come at the cost of overall capability.

C.4 Model Backbone

All experiments are conducted on the **DeepSeek-V2-Lite** model (DeepSeek-AI et al., 2024a) as the backbone. DeepSeek-V2-Lite is a state-of-the-art Mixture-of-Experts Transformer with 26 layers, each containing 66 experts. At each MoE layer, a small subset of experts (8 out of 66) is activated per token based on a learned gating function. This fine-grained expert allocation provides a rich capacity for specialization, making the model ideal for our approach which focuses on expert-specific fine-tuning. We initialize the model from the **public ESFT checkpoint** released by (Wang et al., 2024). This checkpoint comes from a prior alignment training phase in which the model was instruction-tuned on a wide-ranging alignment dataset of conversational and task-following data. Importantly, the alignment data was carefully curated to exclude any math or coding examples. This ensures the base model has strong general alignment (instruction-following and multi-domain conversational skills) without having been specifically trained on math or code problems, providing a neutral starting point to test our specialization methods on those domains.

The aligned base model already demonstrates broad capabilities across many domains (thanks to the alignment phase) while leaving clear room for improvement in mathematical reasoning and coding, as well as providing no unfair advantage on the new tasks (preventing data leakage for math/code evaluations).

C.5 Hyperparameters and Training Settings

We apply a consistent training setup for all compared methods to ensure a fair evaluation. Fine-tuning is performed with a batch size of 32 and a maximum sequence length of 4096 tokens per sample, which accommodates the few-shot context plus the query. For each domain or task fine-tuning, we train for at most 500 steps, which was sufficient for convergence in our experiments. Model performance on a validation set is evaluated every 100 steps to monitor training progress and select the best checkpoint. We conduct a small hyperparameter search for the learning rate in the set {1e-5, 3e-5, 1e-4, 3e-4}, and choose the best learning rate for each method. The resulting learning rates are 3e-5 for Full Fine-Tuning (FFT) (tuning all model parameters), 1e-4 for LoRA (tuning low-rank adapter parameters), 1e-5 for ESFT, 1e-4 for and DES-MoE. Unless otherwise specified, these learning rates are used in all experiments for the respective methods.

Following Wang et al. (2024), for the LoRA method, we use a low-rank adaptation with rank = 8 and a scaling factor = 2, following the configuration in (Hu et al., 2022).

All fine-tuning runs are carried out on high-performance infrastructure: specifically, we use 2 servers each with 8× NVIDIA A100 40GB GPUs (16 GPUs in total), which allows us to accommodate the model's memory needs and train with the full 4096-token context window.

D Evaluation Instructions for Specialized Tasks

Table 4 presents the detailed criteria to evaluate specialized tasks including text summarization, legal judgment prediction, and low-resource translation. Each task includes specific instructions on assessing predicted answers against reference answers, focusing on aspects such as content accuracy, completeness, relevance, and consistency.

Task	Evaluation Instruction
Summary	请你进行以下电话总结内容的评分。请依据以下标准综合考量，以确定预测答案与标准答案之间的一致性程度。满分为10分，根据预测答案的准确性、完整性和相关性逐项扣分。请先给每一项打分并给出总分，再给出打分理由。总分为10分减去每一项扣除分数之和，最低可扣到0分。请以“内容准确性扣x分，详细程序/完整性扣x分，...，总分是：x分”为开头。1. 内容准确性：- 预测答案是否准确反映了客户问题或投诉的核心要点。- 是否有任何关键信息被错误陈述或遗漏。2. 详细程度/完整性：- 预测答案中包含的细节是否充分，能否覆盖标准答案中所有重要点。- 对于任何遗漏的关键信息，应相应减分。3. 内容冗余度：- 预测答案是否简洁明了，和标准答案风格一致，不存在冗余信息。- 如果预测答案过长或与标准答案风格不一致，需相应减分。4. 行为指令正确性：- 预测答案对后续处理的建议或请求是否与标准答案相符。- 如果处理建议发生改变或丢失，需相应减分。预测答案：{prediction} 参考答案：{ground_truth}
Law	请你进行以下法案判决预测内容的评分，请依据以下标准综合考量，以确定预测答案与标准答案之间的一致性程度。满分为10分，根据预测答案的准确性、完整性和相关性来逐项扣分。请先给每一项打分并给出总分，再给出打分理由。总分为10分减去每一项扣除分数之和，最低可扣到0分。请以“相关性扣x分，完整性扣x分，...，总分是：x分”为开头。1. 相关性：预测答案与标准答案的相关程度是最重要的评判标准。如果预测的判决情况与标准答案完全一致，即所有事实和结果都被精确复制或以不同但等效的方式表述，则应给予高分。若只有部分一致或存在偏差，则根据一致的程度适当扣分。如果没有预测判决内容，扣10分。2. 完整性：评估预测答案是否涵盖了所有标准答案中提到的关键点，包括但不限于当事人、具体金额、责任判定、费用承担等。如果遗漏重要信息，则应相应扣分。3. 准确性：检查预测答案中提及的细节、数字、日期和法律依据是否与标准答案保持一致。任何错误信息均需扣分，并且严重错误应该导致更多的扣分。4. 客观性与专业性：预测答案应客观反映法案内容并使用恰当的法律术语。主观臆断或非专业表酌情扣分。预测答案：{prediction} 参考答案：{ground_truth}
Translation	You are an expert master in machine translation. Please score the predicted answer against the standard answer out of 10 points based on the following criteria: Content accuracy: Does the predicted answer accurately reflect the key points of the reference answer? Level of detail/completeness: Does the predicted answer cover all important points from the standard answer? Content redundancy: Is the predicted answer concise and consistent with the style of the standard answer? Respond following the format: "Content accuracy x points, level of detail/completeness x points, total score: x points". The total score is the average of all the scores. Do not give reasons for your scores. Predicted answer: {prediction} Reference answer: {ground_truth}

Table 4: Task instructions for model performance evaluation. The placeholder `{prediction}` and `{ground_truth}` represent model prediction and reference answer, respectively.