Domain-Specific Pruning of Large Mixture-of-Experts Models with Few-shot Demonstrations

Zican Dong[1,2*], **Han Peng**[1,2*], **Peiyu Liu**[3†], **Wayne Xin Zhao**[1,2†], **Dong Wu**[5], **Feng Xiao**[4], **Zhifeng Wang**[4]
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2] Beijing Key Laboratory of Research on Large Models and Intelligent Governance
[3] University of International Business and Economics
[4] YanTron Technology Co. Ltd [5] EBTech Co. Ltd
{dongzican, panospeng}@ruc.edu.cn
liupeiyustu@163.com, batmanfly@gmail.com,
wudong@yantronic.com, {fengx, zhifengw}@ebtech.com

Abstract

Mixture-of-Experts (MoE) models achieve a favorable trade-off between performance and inference efficiency by activating only a subset of experts. However, the memory overhead of storing all experts remains a major limitation, especially in large-scale MoE models such as DeepSeek-R1 (671B). In this study, we investigate domain specialization and expert redundancy in large-scale MoE models and uncover a consistent behavior we term *few-shot expert localization*, with only a few in-domain demonstrations, the model consistently activates a sparse and stable subset of experts on tasks within the same domain. Building on this observation, we propose a simple yet effective pruning framework, **EASY-EP**, that leverages a few domain-specific demonstrations to identify and retain only the most relevant experts. EASY-EP comprises two key components: **output-aware expert importance assessment** and **expert-level token contribution estimation**. The former evaluates the importance of each expert for the current token by considering the gating scores and L2 norm of the outputs of activated experts, while the latter assesses the contribution of tokens based on representation similarities before and after routed experts. Experiments on DeepSeek-R1 and DeepSeek-V3-0324 show that our method can achieve comparable performances and $2.99\times$ throughput under the same memory budget as the full model, with only half the experts. Our code is available at https://github.com/RUCAIBox/EASYEP.

1 Introduction

Mixture-of-Experts (MoE) architectures have been widely adopted as the backbones of various large language models (LLMs) due to their efficiency of scaling parameters without proportional computational overhead [1–4]. However, the deployment of large MoE models imposes substantial memory requirements. Taking DeepSeek-R1 (671B) [1] as an example, it takes about 1500 GB under BF16 precision and 750 GB under FP8 precision, necessitating 4×8 A800 or 2×8 H800 GPU configurations, respectively. This underscores the critical need to explore lite deployment strategies for large-scale MoE models like DeepSeek-R1.

Various training-free approaches have been proposed to alleviate the inference memory demands of MoE models. Expert pruning reduces memory by removing less important experts. Among them,

*Equal Contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-3459

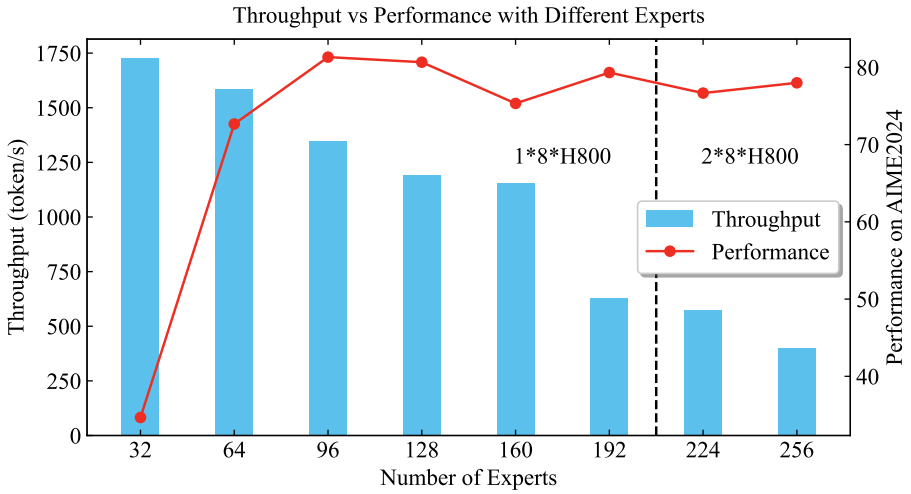

Figure 1: Throughput and performance comparison of DeepSeek-R1 on AIME2024 with varying expert numbers using EASY-EP. We deploy DeepSeek-R1 with two 8× H800 for 224 and 256 experts, while one 8× H800 for others. The throughputs of the latter configurations are multiplied by 2.

router-based methods use expert activation statistics to estimate importance [5], while perturbation-based ones select expert subsets to minimize hidden state drift [6, 7]. However, the former may fail to identify key experts, and the latter is expensive as the number of experts increases. Expert merging combines similar experts to reduce their number [8, 9], but can cause neuron misalignment of experts in MoE models trained from scratch [10, 11]. Moreover, existing methods are primarily designed for MoEs with a few experts per layer (*e.g.,* Mixtral 8×7B [2]) [6]. This highlights the need for more scalable and accurate pruning strategies tailored to large-scale MoE models. While such a scale poses challenges for current approaches, it also brings new opportunities: the increased granularity and domain specialization of experts in models like DeepSeek-R1 [12, 13] make them especially amenable to domain-specific pruning at high compression ratios [14].

In this study, we analyze how expert activations in a large MoE model vary across domains and respond to a small number of demonstration samples. We observe a consistent behavior: given just a few domain-specific examples, the model tends to activate a stable and sparse subset of experts that are highly relevant to the target domain. We refer to this phenomenon as *few-shot expert localization*. To better understand the mechanism behind this behavior, we focus on two factors: the domain specialization of experts and the sufficiency of limited demonstrations. Our key findings are as follows: (1) High gating-value experts are strongly domain-specific, consistently dominating activations within their respective domain while remaining inactive in unrelated ones. (2) A small number of demonstrations suffices to reliably trigger these experts, and they generalize well to other unseen datasets within the same domain.

Based on our observations, we introduce a domain-specific pruning framework that leverages few-shot demonstrations to address memory constraints in large-scale MoE models. Our approach begins by sampling a small number of task demonstrations from a specific domain and generating responses with the original model, serving as a calibration set. To identify and retain the most critical experts, we then develop a pruning method, **E**xpert **A**ssessment with **S**imple **Y**et-effective scoring for **E**xpert **P**runing, *a.k.a.,* **EASY-EP**. This method estimates expert importance and retain a fixed-size subset of top-scoring experts. EASY-EP consists of two complementary components: (1) output-aware expert importance assessment, which combines gating values and the L2 norm of expert outputs to estimate per-token expert importance, and (2) expert-level token contribution estimation, which measures the similarity between the input and the residual-connected output of routed experts to measure each token's contribution to the overall expert score. Notably, the entire scoring process requires only a single forward pass, eliminating the need for backpropagation or repeated evaluations.

To evaluate the efficacy of our approach, we conducted systematic experiments on DeepSeek-R1 and DeepSeek-V3-0324 using eight benchmark datasets covering math, coding, science, finance,

medicine, and agent execution capabilities. Specifically, with only retaining 50% experts, our method can keep comparable performances under the domain-specific pruning settings while achieving over 90% of the full model's performances on DeepSeek-R1 and even better performances on DeepSeek-V3-0324 under the mixed-domain pruning settings. As shown in Figure 1, the performances degrade only slowly with the increase of compression ratio, indicating robustness to pruning. Additionally, under identical memory constraints, pruning 50% of the experts yields a $2.99\times$ increase in inference throughput for sequences of 1K input and 1K output lengths, highlighting the practical utility of our framework in real-world deployments.

2 Background

MoE architectures introduce MoE modules where parameters are dynamically activated to replace feedforward networks (FFNs) [2, 1]. Specially, in the l-th layer, a MoE module contains a router $\mathrm{G}(\cdot)$ and N routed experts $\{\mathrm{E}_1^l(\cdot), \ldots, \mathrm{E}_N^l(\cdot)\}$ [3]. Given an input representation sequence $\mathbf{H}^l = \{\boldsymbol{h}_1^l, \ldots, \boldsymbol{h}_T^l\}, \forall \boldsymbol{h}_t^l \in \mathbb{R}^D$, the router computes the logit of each expert for the t-th token and applies a gating function on the Top-K logits to obtain the gating values $g_{i,t}^l$ (the gating values of deactivated experts $g_{i,t} = 0$). The Top-K experts are activated, and their outputs are aggregated via weighted summation. The final output is obtained by residual connections of the input and output of experts:

$$\bar{\boldsymbol{h}}_t^l = \sum_{i=1}^{N} g_{i,t}^l \cdot \mathrm{E}_i^l(\boldsymbol{h}_t^l), \quad \tilde{\boldsymbol{h}}_t^l = \boldsymbol{h}_t^l + \bar{\boldsymbol{h}}_t^l \tag{1}$$

With the gating values of the router, we define two metrics to assess the importance of each expert, *i.e., frequency* and *gating scores* [5]. For all tokens in a calibration set and an expert E_i^l, we define the frequency f_i^l as the number of times each expert is activated, while the gating scores r_i^l is defined as the total sum of the gating values when each expert is activated, as shown in Equation 2. Here, M is the size of the calibration set, and T_n denotes the number of tokens per demonstration.

$$f_i^l = \sum_{n=1}^{M} \sum_{t=1}^{T_n} (g_{i,n,t}^l > 0), \quad r_i^l = \sum_{n=1}^{M} \sum_{t=1}^{T_n} g_{i,n,t}^l, \tag{2}$$

3 Empirical Analysis of Experts

Previous work has demonstrated that the expert distributions of MoEs with few experts (*e.g.,* Mixtral 8×7B) mainly depend on the syntax structures instead of domains [2]. However, recent large-size MoE models are equipped with various fine-grained experts (*e.g.,* DeepSeek-R1 has 256 experts per layer), which may be more specialized and store distinct knowledge and capacities in their parameters. Motivated by this, we empirically study a phenomenon we term *few-shot expert localization*, where domain-specific experts can be reliably identified using only a handful of demonstrations.

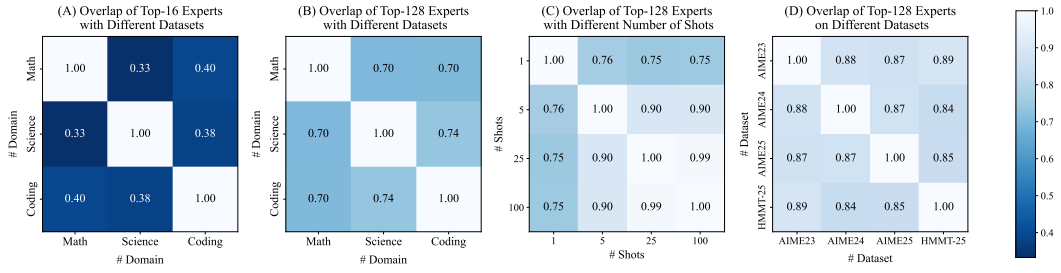

Figure 2: (A), (B): Overlap ratios of experts with top gating scores on different datasets. (C): Overlap ratio of top-128 experts with different numbers of demonstrations. (D): Overlap ratio of top-128 experts pruned with different math datasets.

[3] Shared experts are employed in some MoE models [1, 14–16], but are not considered in this work.

https://doi.org/10.52202/085713-3459

Table 1: Results of removing domain-specific experts. **Bold** denotes in-domain results.

Domain	AIME24	GPQA	LiveCodeBench
Full	77.08	70.91	63.32
Math	67.33 **(-9.75)**	69.19 (-1.72)	65.27 (+1.95)
Code	78.67 (+1.59)	71.72 (+0.81)	**55.68 (-6.64)**
Science	79.33 (+2.25)	**59.09 (-11.82)**	61.07 (-2.25)

3.1 Expert Specialization Across Domains

To assess whether experts in large MoE models exhibit domain-specific specialization, we select AIME-2023 [17], GPQA-main [18], and LiveCodeBench-V3 [19] as calibration datasets for the domains of math, science, and coding, respectively. We conduct experiments using the representative model DeepSeek-R1 on these datasets and extract the gating scores across different domains (as described in Section 2). By analyzing the expert activation distributions across these domains, we aim to reveal how expert utilization varies and assess the degree of specialization.

Distinct Expert Distribution Across Domains. We first rank all experts at each layer by their gating scores and select the top-16 and top-128 experts for each dataset. Subsequently, we measure the overlap of top-ranked experts across different domains and visualize the overlap ratios in Figure 2, where (A) corresponds to top-16 and (B) to top-128. We observe that the top-16 experts are largely disjoint across datasets. When expanding the selection to top-128, the degree of overlap increases, but a significant portion of the experts remains domain-specific. This indicates that *large MoE models contain domain-specialized experts that are predominantly activated in their respective domains.* We show experiments on different numbers of top experts and layer variations in Appendix A.

Impact of Removing Domain-Specific Experts. In order to explore the importance of domain-specific experts, we remove those that appear in the top-128 (by gating score) in a specific domain but not in any others. We then evaluate each pruned model on tasks from the same domain as the calibration data (*i.e.,* in-domain), and on tasks from other domains (*i.e.,* out-of-domain), to assess the generalization behavior of the remaining experts. As shown in Table 1, pruning these experts leads to significant performance degradation on in-domain tasks while having minimal impact on out-of-domain tasks. These results suggest that *domain-specific experts play a critical role in the relevant domain but are redundant for other domains.*

3.2 Expert Locality Within One Domain

Beyond examining the expert specialization across different domains, we examine the locality and stability of expert activation within a single domain. We investigate how the number of demonstrations and the choice of calibration set influence expert selection patterns under the same domain setting.

Effect of Calibration Set Size. Given the presence of domain-specific experts in DeepSeek-R1, a fundamental question arises: How many demonstrations are necessary to accurately identify these key experts? To answer this, we sample varying numbers of demonstrations (*i.e.,* 1, 5, 25, 100) from LiveCodeBench-v3 [19]. For each setting, we compute the average gating scores and retain the top 128 experts. Based on these selections, we then calculate the pairwise overlap ratios between expert sets derived from different demonstration sizes. As illustrated in Figure 2 (C), even with just five demonstrations, over 90% of the critical experts can be effectively identified. Furthermore, 25 demonstrations are sufficient to capture all domain-specific experts (99%), with additional demonstrations yielding only marginal improvements. These results underscore *the feasibility of few-shot domain-specific expert pruning*.

Consistency of Expert Activation Across Datasets. To investigate the consistency of domain-specific experts across datasets within the same domain, we conduct experiments with DeepSeek-R1 on four math datasets: AIME-2023, AIME-2024, AIME-2025, and HMMT-Feb 2025 [17]. Specifically, we perform expert pruning under the 25-shot setting for each dataset and retain Top-128 experts based on their average gating scores. We then compute the pairwise overlap ratios between each pair of datasets. As shown in Figure 2 (D), the overlaps exceed 84% across all math datasets,

revealing a high degree of consistency in domain-specific expert activation. This indicates that *domain-specific expert activation patterns are largely transferable* within the same domain. Despite some dataset-specific differences, the overall expert overlap remains strong and stable.

Empirical analysis with another metric is shown in Appendix B.

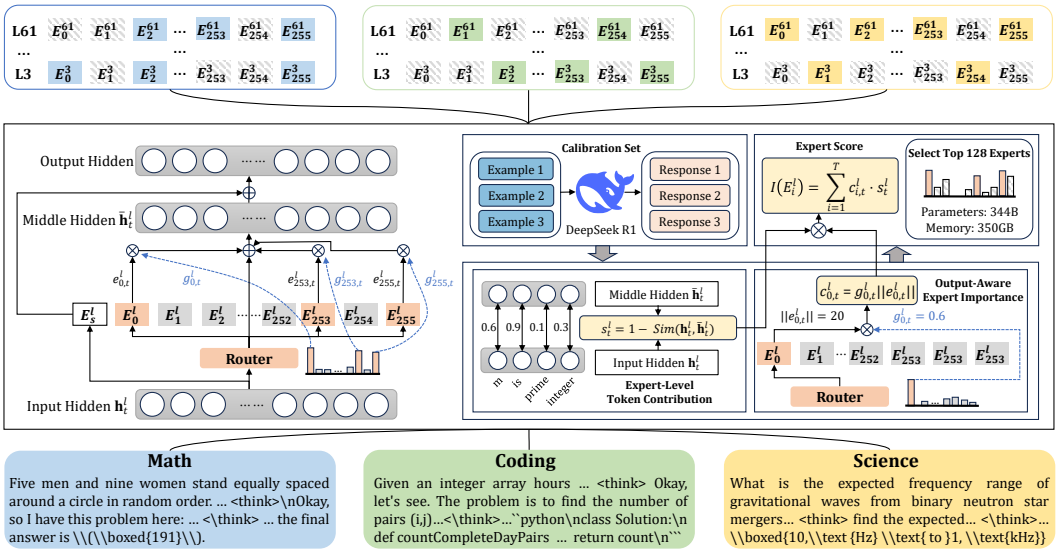

Figure 3: Overall framework of EASY-EP. Given a calibration set consisting of input and responses by the model, EASY-EP leverages output-aware expert importance assessment and expert-level token contribution estimation to compute the expert score on the domain and returns the pruned expert sets.

4 Method

4.1 Overview

Motivated by our earlier observation of few-shot expert localization phenomena in Section 3, we propose an expert pruning framework to reduce memory costs. We use a small set of target-domain demonstrations to run the MoE model and collect expert activation statistics, considering both inputs and outputs for pruning. To effectively identify domain-specific experts in large MoE models, we introduce a simple yet effective expert pruning method, **EASY-EP**. Specifically, we first compute the product of the expert output L2 norm and its corresponding gating value as the output-aware expert importance $c_{i,t}^l$. Next, we determine the expert-level token contribution s_t^l based on the similarity of representations before and after expert computation. The final expert score $I(\mathrm{E}_i^l)$ is obtained by aggregating the product of two terms over all tokens:

$$I(\mathrm{E}_i^l) = \sum_{t=1}^{T} c_{i,t}^l \cdot s_t^l. \tag{3}$$

Our method also supports mixed-domain pruning by averaging the normalized expert scores of all target domains. Thus, we can prune a single model to handle tasks across multiple domains:

$$I_{mix}(\mathrm{E}_i^l) = \sum_{\tau \in \mathcal{T}} (I_\tau(\mathrm{E}_i^l) / \sum_{j=1}^{N} I_\tau(\mathrm{E}_j^l)). \tag{4}$$

Based on the expert scores computed from a small subset of data, we can efficiently select the Top-M experts with the highest scores as retained experts while pruning other experts to reduce memory costs. The overall framework of our method is illustrated in Figure 3.

4.2 Output-Aware Expert Importance Assessment

To assess the importance of each expert, prior router-based expert pruning methods assume activated expert gating scores can reflect their importance [5]. However, this assumption has not considered

https://doi.org/10.52202/085713-3459

the influence of experts. To further assess the contribution of each routed expert, we example the aggregated output $\bar{\boldsymbol{h}}_t^l$ from all routed experts for a given token:

$$\bar{\boldsymbol{h}}_t^l = \sum_{i=1}^{N} g_{i,t}^l \cdot \boldsymbol{e}_{i,t}^l = \sum_{i=1}^{N} g_{i,t}^l \|\boldsymbol{e}_{i,t}^l\| \cdot \frac{\boldsymbol{e}_{i,t}^l}{\|\boldsymbol{e}_{i,t}^l\|}, \tag{5}$$

where $\|\cdot\|$ denotes the L2 norm of a vector, $\boldsymbol{e}_{i,t}^l = \mathrm{E}_i^l(\boldsymbol{h}_t^l)$ represents the output of the expert, and $\frac{\boldsymbol{e}_{i,t}^l}{\|\boldsymbol{e}_{i,t}^l\|}$ denotes the unit vector in the direction of the expert's output. Further, we can compute the upper bound of the L2 norm of the expert outputs as follows:

$$\|\bar{\boldsymbol{h}}_t^l\| \leq \sum_{i=1}^{N} \left\| g_{i,t}^l \|\boldsymbol{e}_{i,t}^l\| \cdot \frac{\boldsymbol{e}_{i,t}^l}{\|\boldsymbol{e}_{i,t}^l\|} \right\| = \sum_{i=1}^{N} g_{i,t}^l \|\boldsymbol{e}_{i,t}^l\|. \tag{6}$$

This indicates that each expert's contribution to the final output is bounded by the product of its gating value and the L2 norm of its output, $g_{i,t}^l \|\boldsymbol{e}_{i,t}^l\|$. An expert with a large gating score may still produce outputs with low L2 norm, ultimately resulting in a limited influence on the final output, which are empirically verified in Appendix C.1. Therefore, instead of using only the gating value, we define the importance of an expert for a given token as the product of its gating value and L2 norm of output, as formalized in the following equation:

$$c_{i,t}^l = g_{i,t}^l \|\boldsymbol{e}_{i,t}^l\|, \quad \forall g_{i,t}^l > 0. \tag{7}$$

4.3 Expert-Level Token Contribution Estimation

When calculating the statistical metrics for evaluating the importance of experts, prior work often directly averages the scores across all tokens [6, 5]. However, in practice, the influence of routed experts' outputs on the residual stream varies significantly across tokens (as shown in Appendix C.2). Intuitively, when dealing with tokens exhibiting low similarity before and after the MoE module, adjusting their routed experts will induce a substantial distributional shift in their representations. In contrast, for tokens with high similarity, such adjustments will lead to only minimal drift in their representational distributions [20, 21]. Inspired by these, we propose a similarity-based token importance assessment method which give greater weights for the former tokens. Given the representations before and after the routed expert modules $\boldsymbol{h}_t^l$ and $\tilde{\boldsymbol{h}}_t^l$, we compute the cosine similarity between these representations. The token importance score s_t^l is then defined as one minus this similarity, capturing the extent of change induced by the routed expert module:

$$s_t^l = 1 - \mathrm{Sim}(\boldsymbol{h}_t^l, \tilde{\boldsymbol{h}}_t^l). \tag{8}$$

5 Experiments

5.1 Experimental Settings

Evaluation Benchmarks. To systematically assess the effectiveness of our proposed method, we conduct experiments across eight benchmark datasets: AIME-2024, AIME-2025, HMMT-Feb 2025, LiveCodeBench [19], GPQA-Diamond [18], USMLE [22], FinanceIQ [23], and AgentBench-OS [24]. These benchmarks encompass six fundamental domains and tasks of LLMs: math, coding, science, medicine, finance, and agent-based task execution.

Experiment Settings. We select DeepSeek-R1 [1] and DeepSeek-V3-0324 [15] as our evaluated models and consider domain-specific and mixed-domain pruning settings. For each domain, we randomly sample 25 instances and construct a calibration set by concatenating their inputs with the target model's outputs. We then evaluate the expert scores on the calibration data and select the top 64 and 128 experts with the highest scores at each layer, respectively. We also average the normalized expert scores on different domains to evaluate the mixed-domain pruning performances. Details regarding the candidate sets and evaluation settings are provided in Appendix D.

Table 2: Comparison of the performances of different expert pruning methods. HMMT denotes HMMT-Feb 2025, GPQA denotes GPQA-Diamond, A-OS denotes AgentBench-OS, and FinIQ denotes FinanceIQ.

Model	Method	Mix	#E	AIME-24	AIME-25	FMMT	LiveCode	GPQA	USMLE	FinIQ	A-OS	Avg
DeepSeek-R1	Full	-	256	77.08	66.67	44.38	63.32	70.91	92.66	82.1	40.51	67.20
	Random	×	64	0.00	0.00	0.00	0.00	26.09	0.00	0.00	0.00	3.26
	Frequency	×	64	0.00	0.00	0.00	0.00	17.68	0.00	0.00	2.78	2.58
	Gating Score	×	64	2.67	1.33	2.67	14.97	46.83	0.86	0.00	0.69	8.75
	M-SMoE	×	64	0.00	0.00	0.00	0.00	12.12	0.00	0.00	0.00	1.52
	EASY-EP	×	64	**72.81**	**55.10**	**38.02**	**42.51**	**67.47**	**26.63**	**33.90**	**27.26**	**45.22**
	Random	×	128	8.33	6.67	3.33	20.96	34.95	57.66	0.00	7.64	17.44
	Frequency	×	128	19.33	13.33	7.33	36.08	59.60	61.51	26.40	29.16	31.59
	Gating Score	×	128	70.10	55.52	36.15	47.60	63.78	80.36	66.50	31.94	56.49
	M-SMoE	×	128	5.33	6.00	3.33	25.75	24.75	52.63	39.60	19.44	22.10
	EASY-EP	×	128	**79.17**	**68.33**	**45.31**	**61.11**	**70.12**	**91.67**	**78.80**	**37.92**	**66.55**
	Frequency	✓	128	21.33	10.00	6.00	7.49	41.45	78.55	**62.14**	11.81	29.85
	Gating Score	✓	128	29.33	21.33	18.00	22.75	41.69	62.06	27.29	30.56	31.67
	M-SMoE	✓	128	6.67	2.00	4.67	4.19	32.32	72.00	19.10	6.25	18.40
	EASY-EP	✓	128	**75.94**	**61.98**	**42.50**	**57.63**	**70.36**	**91.20**	57.95	**34.17**	**61.47**
DeepSeek-V3-0324	Full	-	256	55.73	47.71	28.75	48.50	66.87	87.51	64.22	33.33	54.08
	Random	×	64	0.00	0.00	0.00	0.00	26.87	0.39	0.00	0.69	3.49
	Frequency	×	64	31.35	34.06	15.73	1.95	45.25	40.13	61.96	22.74	31.65
	Gating Score	×	64	43.96	25.10	23.12	14.97	51.52	78.68	64.20	0.00	37.69
	M-SMoE	×	64	16.67	13.33	3.33	1.20	22.22	12.18	47.00	21.52	17.18
	EASY-EP	×	64	**53.12**	**41.56**	**28.85**	**27.99**	**57.35**	**84.57**	**72.50**	**27.55**	**49.19**
	Random	×	128	1.33	0.67	0.00	11.38	34.95	53.5	53.66	18.75	21.78
	Frequency	×	128	**55.73**	42.60	30.10	36.08	63.54	84.29	66.84	31.71	51.36
	Gating Score	×	128	55.42	45.10	30.94	**47.60**	63.78	84.62	**67.76**	35.42	53.83
	M-SMoE	×	128	48.00	38.67	28.67	30.53	55.82	86.72	66.60	33.33	48.54
	EASY-EP	×	128	55.21	**46.88**	**31.56**	46.71	**65.25**	**86.72**	63.58	**37.08**	**54.12**
	Frequency	✓	128	51.35	37.60	24.27	17.07	55.90	83.47	66.80	36.25	46.59
	Gating Score	✓	128	53.75	40.10	27.19	28.74	58.88	83.86	67.74	34.58	49.36
	M-SMoE	✓	128	43.33	30.00	20.00	7.19	52.53	82.33	62.20	29.17	40.84
	EASY-EP	✓	128	**57.81**	**46.56**	**33.33**	**40.72**	**64.95**	**85.00**	**72.26**	**38.74**	**54.92**

Baselines. In our experiments, we employ three expert pruning methods and one expert merging method for comparison. For expert pruning, we employ different methods to assess the expert scores, including *random*, *frequency*, and *gating scores* (as discussed in Equation 2), and only keep Top-M experts with the highest expert scores [4]. For expert merging, we select M-SMoE [25], which first employs neuron permutation alignment to mitigate neuron misalignment of experts and then merges experts into dominant ones with similarity of router logits.

5.2 Main Results

Table 2 showcases our method's performance against baselines under various pruning configurations. First, our approach consistently outperforms all baseline methods across diverse benchmarks and pruning settings. Notably, in domain-specific pruning, our method matches and even surpasses full model performance on certain benchmarks with only half the experts. This may be attributed to the effective removal of irrelevant experts, enhancing the model's ability to utilize domain-specific knowledge. Furthermore, our method demonstrates strong resilience to high compression ratios (*e.g.,* 75%), where most methods experience significant performance degradation, particularly on DeepSeek-R1. In contrast, our technique preserves substantial model capabilities, highlighting its effectiveness in identifying critical experts for specific tasks.

Second, non-reasoning models exhibit greater robustness compared to their reasoning-oriented counterparts. DeepSeek-V3-0324, for instance, retained more performance across most pruning methods than DeepSeek-R1. Under domain-specific pruning, DeepSeek-V3-0324's performance on datasets like AgentBench-OS even improved notably after pruning some experts. However, this phenomenon is not observed with DeepSeek-R1. We hypothesize that while domain capabilities might be preserved in pruned reasoning models, their long-term generation abilities are compromised.

[4]We do not include perturbation-based pruning methods in our comparison [6, 7], which is computationally prohibitive for MoE models with 256 experts (the detailed analysis is shown in Appendix E).

 https://doi.org/10.52202/085713-3459

Finally, our method also excels in preserving performance under mixed-domain pruning. It retains over 90% of the original performance, surpassing domain-specific compression with other methods. Conversely, other expert pruning techniques struggle to maintain balanced performance across different domains. This underscores that the overlapped experts identified by our approach, which are linked to general reasoning abilities, effectively contribute to a broad array of downstream domains.

5.3 Detailed Analysis

In this section, we conduct further ablation studies and detailed analyses to investigate the effectiveness of our approach and the few-shot expert localization phenomena of large MoE models. For more analysis experiments, we present them in Appendix F.

5.3.1 Ablation Study

We conduct ablation studies to analyze the impact of each component in our method. Specifically, we evaluate two variants with 64 remaining experts: (1) removing the token-level contribution estimation and (2) replacing the product of gating values and L2 norm of expert outputs with only the gating scores. As shown in Table 3, both incorporating token-level contribution estimation and considering the L2 norm of expert outputs lead to improved performance compared to using only the gating score. Furthermore, combining both components results in the best overall performance. These findings highlight the importance of both components in our method.

Table 3: Results of ablation study. norm denotes whether considering L2 norm of expert output and Token denotes whether considering token contribution scores.

Method	Metric	Experts	AIME-24	AIME-25	HMMT	LiveCode	GPQA	A-OS
Ours	$g^l_{i,t}\|\boldsymbol{e}^l_{i,t}\| \cdot s^l_t$	64	72.81	55.33	36.00	42.51	67.47	27.26
w/o Token	$g^l_{i,t}\|\boldsymbol{e}^l_{i,t}\|$	64	65.33	49.33	31.33	27.54	56.57	21.53
w/o norm	$g^l_{i,t} \cdot s^l_t$	64	70.00	40.00	23.33	19.76	61.11	18.75
w/o both	$g^l_{i,t}$	64	2.67	1.33	2.67	0.00	20.20	0.69

5.3.2 Generalization Capacity

Beyond same-domain task evaluations, we assessed the generalization capacities on unrelated domains after domain-specific pruning (e.g., pruning DeepSeek-R1 with AIME2023 and evaluating on LiveCodeBench). As Table 4 shows, the model demonstrates a certain generalization ability, especially in similar domains (*e.g.,* math/science, code/OS-agent, science/medicine). We hypothesize that while some domain-specific experts are pruned, core reasoning experts are preserved. However, the out-of-domain performances are typically poorer than mixed-domain pruning. Thus, we suggest using mixed-domain pruning when facing multiple downstream tasks.

Table 4: Results of generalization capacities of pruned models. Domain denotes the domain of pruning data. **Bold** denotes in-domain performances.

Domain	AIME24	LiveCodeBench	GPQA	Agent-OS	USMLE	FinIQ
Math	**79.17**	46.11	46.91	3.47	46.43	58.20
Coding	38.00	**61.11**	39.90	15.97	41.79	53.00
Science	64.64	53.59	**70.12**	4.17	75.88	57.50

5.3.3 Effect of Pruning Data

We also study the impact of pruning data. Instead of using the full model's input and output, we examine five types of data: (1) *Inp*: just input context data; (2) *Out*: only the model's generated data; (3) *Inp+Ans*: input context data and the correct generated answer; (4) *PT*: pre-training data from the same domain; and (5) *CC*: data from CommonCrawl. Table 5 and 6 show the experimental results for DeekSeek-R1 and DeepSeek-V3-0324. Compared to using the combination of input and model output, performance decreased under other settings for both math and coding tasks. This suggests that,

even within the same domain, there are notable differences in expert distributions among data types. Additionally, employing CommonCrawl data causes significant performance declines, demonstrating that pruning a task-irrelevant model with pre-training data is not suitable for current models.

Table 5: Comparison of performances of DeepSeek-R1 with different data.

Data	AIME24	LiveCodebench	GPQA
Inp+Out	79.17	61.11	70.12
Inp+Ans	75.33	53.89	67.98
Inp	66.00	50.90	70.81
Out	77.33	59.28	70.10
PT	29.33	38.32	66.16
CC	0.00	0.00	28.92

Table 6: Comparison of performances of DeepSeek-V3-0324 with different data.

Data	AIME24	LiveCodebench	GPQA
Inp+Out	55.73	48.50	66.87
Inp+Ans	54.00	43.11	61.62
Inp	51.33	14.97	63.64
Out	54.67	47.90	63.13
PT	13.33	37.72	56.57
CC	0.00	0.00	29.80

5.3.4 Effect of Number of Pruning Demonstrations

In Section 3.2, we observe that domain-relevant experts can be identified with only a few demonstrations. Here, we further investigate the impact of the number of demonstrations on final performance. To do so, we sample varying numbers of demonstrations from the same distribution and prune half of the experts in each layer. The performance variations on AIME24 and LiveCodeBench are presented in Figure 4. When we utilize only a single sample for pruning, the selected experts are often influenced by the characteristics of that individual sample, thus resulting in lower performance. As we further increase the number of demonstrations, the performance rapidly rises, achieving comparable performances with the full model.

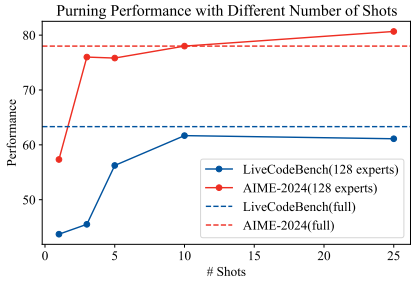

Figure 4: Comparison of performance with different numbers of shots for pruning.

5.3.5 Analysis of Throughput

To evaluate the throughput of pruned models with different numbers of experts, we use the SGLang [26] package and measure performance under a maximum request concurrency of 32. We evaluate the settings with 1K input and 1K output length. For configurations with more than 192 experts, two 8×H800 GPUs are used. Figure 1 shows the scaling throughput for this setting. We observe that reducing the number of experts significantly improves throughput, particularly when the model can be deployed on a single node. Compared to the full DeepSeek-R1 model, configurations with 128 and 64 experts achieve 2.99× and 4.33× throughput, respectively. Compared to the full model, the pruned model can be deployed on a single node, thereby avoiding inter-node communication overhead. Moreover, using fewer experts further reduces communication between GPUs within the node, improving computational efficiency. We provide more experiments in Appendix F.5.

6 Related Work

MoE architectures improve computational efficiency by activating only a small subset of experts per input [27, 28, 4]. However, the increased number of parameters introduced by these architectures leads to substantial memory overhead. To address this, existing efforts broadly fall into two categories. The first line of work focuses on architectural optimization to reduce computation or parameter size. Representative methods include pyramid-shaped expert allocation [29], fine-grained expert design with smaller per-expert modules [14], and the transformation of dense models into sparse MoE variants [30–32]. While effective, these approaches typically require modifying model architecture and retraining from scratch, which limits their applicability to already deployed or pretrained models. The second line of research approaches the problem from a memory efficiency perspective by applying post-hoc compression techniques, primarily pruning and quantization [33–35]. These

https://doi.org/10.52202/085713-3459

methods typically estimate expert importance based on routing frequency [5], gating scores [36], or direct measurement of their contribution to model outputs [6, 7]. Some approaches further reduce redundancy by merging similar experts [8, 9], though they may meet the problem of neuron misalignment and additional memory costs [10, 11]. In contrast, our method enables efficient pruning with a single forward pass and avoids storing additional model variants during compression.

7 Conclusion

In this work, we investigated the domain specialization of experts in large MoE models. Our observations indicate that domain-specific experts play a crucial role in their respective domains and can be effectively identified with a few demonstrations. Building on these insights, we proposed a pruning strategy that leverages demonstrations from tasks within the same domain. Specifically, we introduced EASY-EP, a simple yet effective pruning method that combines output-aware expert importance assessment with expert-level token contribution estimation. Experimental results showed that our approach maintained comparable performance while utilizing only half of the experts in domain-specific settings and retained over 90% of the original performance in mixed-domain pruning. We believe that our method can facilitate the deployment of large MoE models, particularly for efficiently handling a high volume of samples within the same domain.

8 Limitation

Our work investigates the phenomenon of few-shot expert localization in large MoE models and proposes a simple yet effective method for domain-specific expert pruning. We leave the investigation of training to further enhance the pruned model's performance, particularly in balancing in-domain and out-of-domain capabilities, as future work, given our current focus on evaluating pruning effectiveness under realistic resource constraints. We observed differing levels of robustness to expert pruning between reasoning-oriented (DeepSeek-R1) and non-reasoning models (DeepSeek-V3-0324). While this trend is noteworthy, further validation on a broader range of architectures is needed to strengthen the generality of this observation.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205 and 62506077, Beijing Natural Science Foundation under Grant No. L233008 and Beijing Municipal Science and Technology Project under Grant No. Z231100010323009. Peiyu Liu and Xin Zhao are the corresponding authors.

References

[1] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[2] 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. *CoRR*, abs/2401.04088, 2024.

[3] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *CoRR*, abs/2303.18223, 2023.

[4] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *CoRR*, abs/2407.06204, 2024.

[5] Alexandre Muzio, Alex Sun, and Churan He. Seer-moe: Sparse expert efficiency through regularization for mixture-of-experts. *CoRR*, abs/2404.05089, 2024.

[6] 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6159–6172. Association for Computational Linguistics, 2024.

[7] Mingyu Cao, Gen Li, Jie Ji, Jiaqi Zhang, Xiaolong Ma, Shiwei Liu, and Lu Yin. Condense, don't just prune: Enhancing efficiency and performance in moe layer pruning. *CoRR*, abs/2412.00069, 2024.

[8] Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient smoe with hints from its routing policy. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[9] I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse mixture-of-experts via hierarchical clustering. *CoRR*, abs/2410.08589, 2024.

[10] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[11] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the symmetries of deep learning models and their internal representations. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[12] Matthew Lyle Olson, Neale Ratzlaff, Musashi Hinck, Man Luo, Sungduk Yu, Chendi Xue, and Vasudev Lal. Semantic specialization in moe appears with scale: A study of deepseek R1 expert specialization. *CoRR*, abs/2502.10928, 2025.

[13] Robert Dahlke, Henrik Klagges, Dan Zecha, Benjamin Merkel, Sven Rohr, and Fabian Klemm. Mixture of tunable experts–behavior modification of deepseek-r1 at inference time. *arXiv preprint arXiv:2502.11096*, 2025.

[14] 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. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1280–1297. Association for Computational Linguistics, 2024.

 https://doi.org/10.52202/085713-3459

[15] 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, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024.

[16] 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, Guowei Li, Hao Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, Tao Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, and Xiaowen Sun. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434, 2024.

[17] Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025.

[18] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023.

[19] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *CoRR*, abs/2403.07974, 2024.

[20] Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *CoRR*, abs/2403.03853, 2024.

[21] Zican Dong, Junyi Li, Jinhao Jiang, Mingyu Xu, Xin Zhao, Bingning Wang, and Weipeng Chen. LongReD: Mitigating short-text degradation of long-context large language models via restoration distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10687–10707, Vienna, Austria, July 2025.

[22] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081, 2020.

[23] Xuanyu Zhang and Qing Yang. Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 4435–4439. ACM, 2023.

[24] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[25] Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient smoe with hints from its routing policy. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[26] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems*, 37:62557–62583, 2024.

[27] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR (Poster)*. OpenReview.net, 2017.

[28] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

[29] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346. PMLR, 2022.

[30] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 877–890. Association for Computational Linguistics, 2022.

[31] Hongyu Wang, Shuming Ma, Ruiping Wang, and Furu Wei. Q-sparse: All large language models can be fully sparsely-activated. *CoRR*, abs/2407.10969, 2024.

[32] Peiyu Liu, Tianwen Wei, Bo Zhu, Xin Zhao, and Shuicheng Yan. Masks can be learned as an alternative to experts. In *ACL (1)*, pages 15800–15811. Association for Computational Linguistics, 2025.

[33] Peiyu Liu, Zikang Liu, Ze-Feng Gao, Dawei Gao, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Do emergent abilities exist in quantized large language models: An empirical study. In *LREC/COLING*, pages 5174–5190. ELRA and ICCL, 2024.

[34] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[35] Haokun Lin, Haoli Bai, Zhili Liu, Lu Hou, Muyi Sun, Linqi Song, Ying Wei, and Zhenan Sun. Mope-clip: Structured pruning for efficient vision-language models with module-wise pruning error metric. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27370–27380, 2024.

[36] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *ICLR*. OpenReview.net, 2024.

 https://doi.org/10.52202/085713-3459

A Expert Overlap Across Domains

To further analyze the overlap of top experts across different domains, we select AIME-2023, LiveCodeBench-V3, and GPQA as calibration sets and select top experts with gating scores. Subsequently, we compute the overlap ratio of Top-M experts ($M \in \{1, 2, 4, 8, 16, 32, 64, 128\}$) and expert overlap across different layers, which are shown in Figure 5. We can observe that the experts with the highest gating scores differ significantly across different domains, and as the number of top experts increases, the overlap ratio increases. Additionally, in the lower layers, there is a higher overlap among experts with significant gating scores across different datasets, but this overlap becomes relatively lower in the middle and deeper layers. This indicates that experts in the lower layers tend to focus on general capacities, while as the network depth increases, they gradually specialize in handling knowledge from distinct domains.

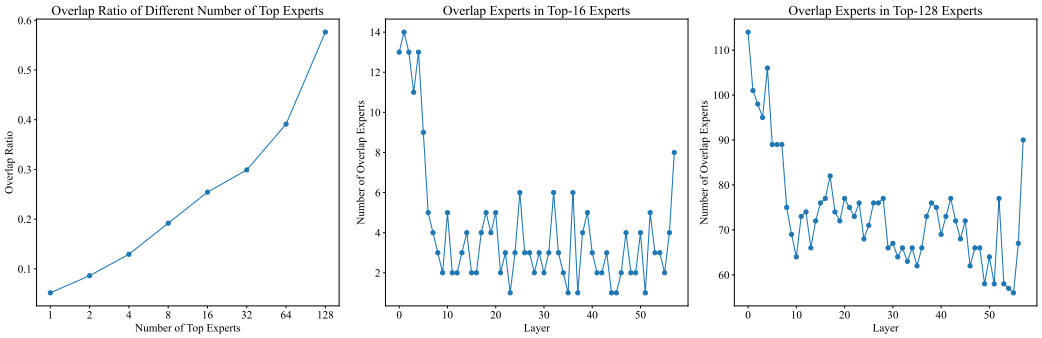

Figure 5: Left: Overlap ratios of Top-M experts with different M. Middle and Right: Overlap experts in top-16 and 128 experts on different layers.

B Empirical Study with EASY-EP

To further illustrate the phenomenon of few-shot expert localization, we analyze it with the expert scores (as discussed in Equation 3) from our proposed EASY-EP method in place of the gating scores.

B.1 Expert Specialization Across Domains

Distinct Expert Distribution Across Domains. We compute the overlap ratios of the top experts identified by our method and visualize them in Figure 6. Similar to previous observations, large differences in expert distribution still exist across different domains. However, the expert overlap between domains is larger than that of experts identified by gating scores. This may be because our method identifies experts not only frequently activated within a domain but also those who contribute more than other experts in residual connections. The latter may be more similar across different datasets, thus leading to higher overlap.

Impact of Removing Domain-Specific Experts. Similarly, we remove experts whose score, as calculated by our method, falls within the top-128 for a single domain but not for any other. As shown in Table 7, removing these in-domain experts leads to larger performance degradation than out-of-domain experts. However, compared with pruning with gating scores, the performance degradation is less severe. We speculate this is due to the higher overlap among the top experts identified by our method, resulting in a smaller pruning ratio.

B.2 Expert Locality Within One Domain

Effect of Calibration Set Size. We present the overlap ratios of top-128 experts selected via our method with different numbers of demonstrations. As shown in Figure 7 (Left), as the number of demonstrations increases, the identified domain-specific experts gradually become stable. For 25-shot

https://doi.org/10.52202/085713-3459

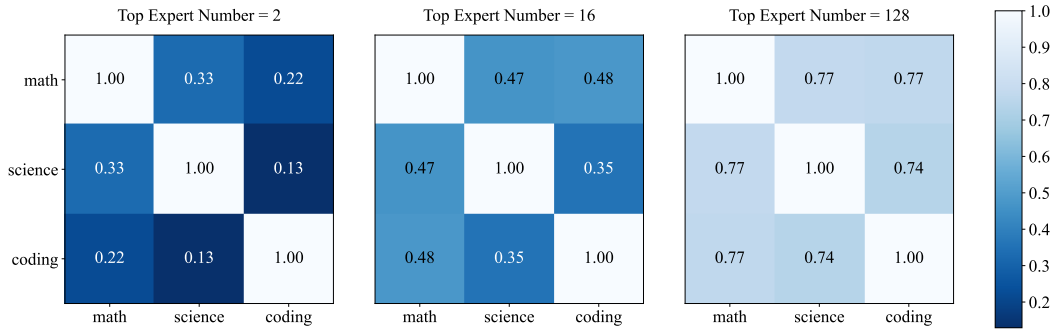

Figure 6: Overlap ratios of experts with top expert scores computed with our methods on different datasets.

Table 7: Results of removing domain-specific experts with EASY-EP.

Domain	AIME24	GPQA	LiveCodeBench
Full	77.08	70.91	63.32
Math	72.67 (**-3.41**)	73.17 (+2.26)	64.22 (+0.90)
Code	76.67 (-0.41)	72.93 (+2.02)	62.27 (**-1.05**)
Science	76.00 (-1.08)	58.23 (**-12.68**)	64.97 (+1.65)

pruning, it can achieve the overlap ratio of 93% with 10-shot pruning, further demonstrating the feasibility of few-shot domain-specific expert pruning.

Consistency of Expert Activation Across Datasets. Figure 7 (Right) illustrates that the top-128 experts identified by our method exhibit consistent overlap ratios across different datasets within the math domain. Our method not only presents expert consistency over different datasets similar to the gating score metric, but also yields significantly higher overlap ratios. We hypothesize that our approach is more adept at identifying experts of true domain-wide significance, as opposed to those who are only important in a single dataset due to the dataset-specific differences.

Figure 7: Left: Overlap ratio of top-128 experts identified by EASY-EP with different numbers of demonstrations. Right: Overlap ratio of top-128 experts identified by EASY-EP pruned with different math datasets.

https://doi.org/10.52202/085713-3459

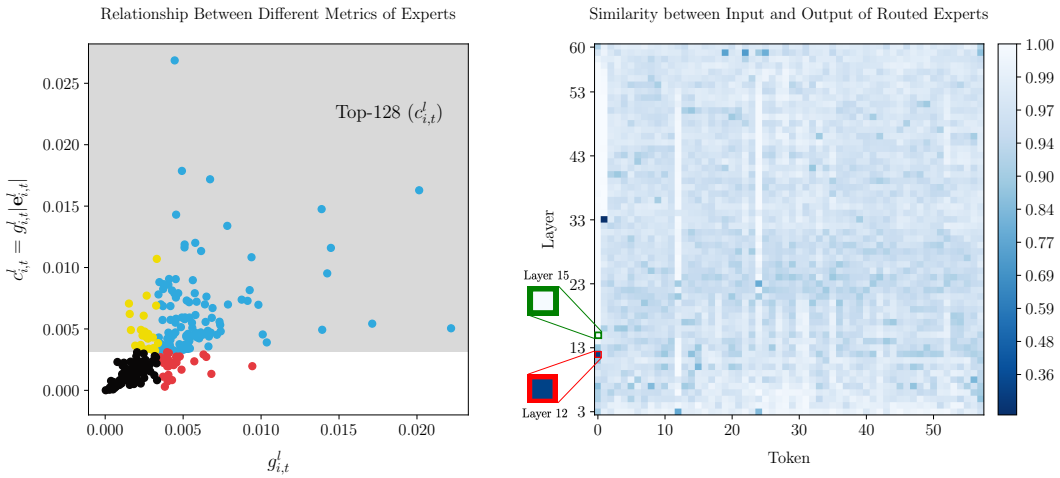

Figure 8: Left: Gating scores and averaged product of gating value and L2 Norm of expert outputs. Blue/red dots indicate experts in top-128 gating scores $g^l_{i,t}$; blue/yellow dots denote experts in top-128 expert importance $c^l_{i,t}$; black dots indicate neither. Right: Consine similarity between representations before and after incorporating the outputs of the routed expert. The red box and green box indicate high similarity and low similarity, respectively.

C Empirical Analysis of Components in EASY-EP

C.1 Output-Aware Expert Importance Assessment

To analyze the relationship between whether a large gating score $g^l_{i,t}$ ensures a large product of gating scores and L2 norm of expert output $g^l_{i,t}\|e^l_{i,t}\|$, we visualize the relationship of top-128 experts selected by the two metrics. As shown in Figure 8 (Left), despite a significant degree of overlap among experts, there still exist some experts who excel exclusively in a single, focused metric. Although the gating scores of some experts are not large, the L2 norms of their outputs are larger than others. This proves the necessity of considering experts' outputs.

C.2 Expert-Level Token Contribution Estimation

Given one sample selected from AIME-2023, we first obtain representations before and after incorporating the outputs of routed experts in each layer of DeepSeek-R1 and compute the similarities between them, as illustrated in Figure 8 (Right). We can observe that the similarities differ significantly across tokens and layers. For some tokens at specific layers, skipping the expert module results in minimal changes to the hidden states, with over 99% similarity between the input and output representations (*e.g.,* the 15th layer of the first tokens). In contrast, for certain tokens and layers (*e.g.,* the 12th layer of the first tokens), the hidden states show substantial differences after expert routing.

D Experiment Details

As shown in the previous analysis in Section 3.2, datasets within the same domain can identify the important experts on other datasets. Thus, for different domains and tasks, we select one dataset as a calibration set, which is shown in Table 8. For mixed-domain pruning, we choose the scores calculated on each 25-shot calibration set and average the normalized scores as the final scores of experts. For the evaluation of FinanceIQ, we randomly select 1000 samples with the seed of 42 since the test set is too large.

Additionally, all the experiments are conducted in one $8\times$ H200 GPU. We set the maximum context length to 32K, the temperature to 0.6, and the top-p sampling value to 0.95 for most benchmarks (temperature as 0.2 for LiveCodeBench). To ensure statistical reliability, most benchmark is evaluated independently 5 times (32 times for math benchmarks), and we report the average performance of pass@1.

Table 8: Calibration Set of Each Domain

Domain	Calibration Set	License
Math	AIME 2023	cc-by-nc-sa-4.0
Coding	LiveCodeBench-V3	MIT
Science	GPQA-Main	MIT
Agent	Dev Set of AgentBench-OS	Apache-2.0 license
Finance	Dev Set of FinanceIQ	MIT
Medical	Dev Set of USMLE	cc-by-nc-sa-4.0

E Analysis of Perturbation-based Pruning Method

Previous studies [6, 7] have proposed methods that utilize representation perturbation after expert pruning to determine which experts should be removed. In NAEE [6], identifying the optimal subset of experts requires $C_N^{N'}$ evaluations, where N and N' represent the original and target numbers of experts, respectively. In CD-MoE [7], a greedy search algorithm selects the expert to retain based on minimal representation perturbation through a rolling mechanism, requiring $N(N+N')/2$ evaluations. For DeepSeek-R1, which has 256 experts and a target expert count of 128, these methods require over 10^{75} and 24768 evaluations per layer, respectively. Thus, the perturbation-based methods are not affordable for MoE models with a large number of experts. Conversely, our method only requires one forward operation to identify critical experts, which is cheaper and more suitable for large MoE models.

F Further Experimental Analysis

F.1 Experiments with Qwen3-30B-A3B

To further demonstrate the effectiveness of our method, we evaluate the performance of Qwen3-30B-A3B with different expert pruning methods. Specifically, the expert number of the model is pruned from 128 to 64. As shown in Table 9, our method can achieve significantly better performances than other methods, which demonstrates the effectiveness of our method. Additionally, compared with the DeepSeek series, it is harder to compress Qwen3-30B-A3B since it has no shared experts.

Qwen-30B-A3B	AIME24	AIEM25	HMMT	GPQA	LiveCodeBench	Averge
FULL	80.42	70.83	50.00	68.59	62.80	66.53
Random	0.00	0.00	0.00	23.23	0.00	4.65
frequency	13.33	10.00	0.00	61.11	14.00	19.69
Gating Scores	13.33	6.67	6.67	35.35	14.00	15.20
EASY-EP	72.92	61.25	38.75	63.13	46.30	56.47

Table 9: Experiments on Qwen3-30B-A3B.

F.2 Expert Overlap with Different Metrics

We calculate the pairwise overlap of the top-128 experts chosen by various metrics across three datasets: AIME23, GPQA-main, and LiveCodeBench-V3. The results are presented as a heatmap in Figure 9. Our analysis reveals that the experts selected based on gating scores and frequency exhibit a remarkably high overlap, approximately 90%. However, the experts identified by our method show significant divergence from these metrics, underscoring the influence of incorporating expert outputs and residual changes in our approach. We posit that the large discrepancy among experts results in a huge performance gap between models.

https://doi.org/10.52202/085713-3459

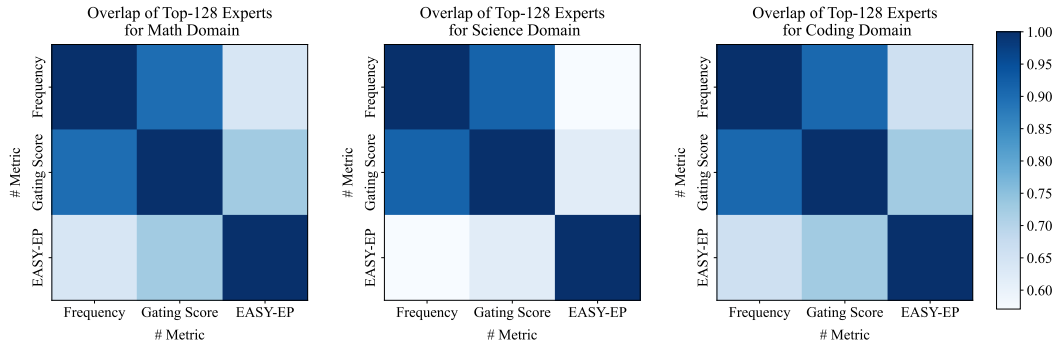

Figure 9: Overlap of Top-128 Experts Selected via Different Metrics.

Model	AIME24	AIME25
FULL	80.42	70.83
EASYEP	72.92	61.25
EASYEP w.o. reroute	73.33	63.33

Table 10: Performances of removing experts without rerouting.

F.3 Effect of Directly Removing Experts Without Reroute

In addition to our settings, where we remain the same number of activated experts, we also experimented with computing the gating weights for each expert as normal and selecting them accordingly. Then, for the pruned experts, we set their outputs to zero. Thus, to some extent, tokens that would have been routed to the pruned experts are processed by fewer experts. We employ Qwen3-30B-A3B for experiments and compare the performances with and without reroute, as shown in Table 10. We can observe that the performances slightly improve compared to the original version. In addition, it can save computation due to the reduced number of activated experts, which can further accelerate the inference speed.

F.4 Results of Pruning with Different Ratios at Different Layers

To further investigate the pruning performance of our method, we employed layer-wise dynamic pruning. Specifically, we first normalized the expert scores for each layer and ranked all experts across all layers. Subsequently, we pruned the top experts using different pruning ratios. As shown in Table 11, we observe that employing layer-wise dynamic pruning does not always lead to better performance, and the performance on LiveCodeBench even drops significantly. In addition, the dynamic compression ratio leads to deployment difficulties, as it requires different numbers of experts for each layer. Thus, we suggest just employing a fixed pruning ratio across all layers.

Table 11: Performance comparison of EASY-EP with and without employing layer-wise compression ratios.

Model	Ratio	Layer	AIME2024	GPQA	LiveCodeBench
DeepSeek-R1	0.5	×	79.17	70.12	61.11
	0.5	✓	79.33	65.15	44.31
	0.25	×	72.67	67.47	42.51
	0.25	✓	65.33	66.16	28.74
DeepSeek-V3-0324	0.5	×	55.21	65.25	46.71
	0.5	✓	58.67	62.63	38.23
	0.25	×	53.12	57.35	27.99
	0.25	✓	58.67	60.10	21.56

F.5 Throughput with Different Lengths

To further explore the throughput changes on pruning models, we consider three additional length settings: (1) *1K+4K*: 1K input length and 4K output length; (2) *4K+1K*: 4K input length and 1K output length; and (3) *4K+4K*: 4K input length and 4K output length. Figure 10 (Right) presents results for the other settings. Compared with *1K+1K*, either increasing the length of input or output leads to lower throughput and throughput acceleration after pruning. Additionally, in long output scenarios with an equivalent total sequence length, the 128-expert configuration achieves a significantly higher acceleration ratio (2.91× under the *1K+4K* setting, compared to 2.52× under the *4K+1K* setting).

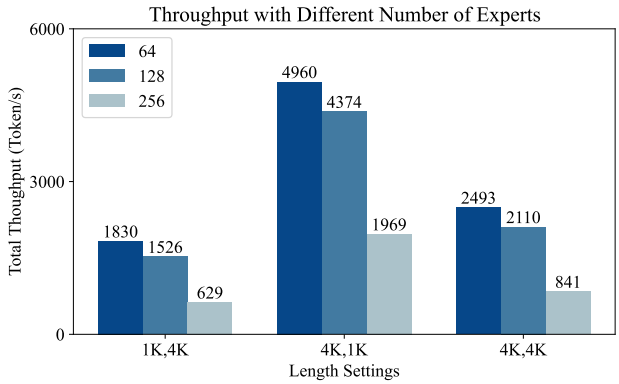

Figure 10: Total throughput across different numbers of experts.

https://doi.org/10.52202/085713-3459

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide the main claims in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include Limitation section in Appendix A.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We have no aussmptions and proof of theoretical result.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the information to reproduce the results in Section 5.1 and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

 https://doi.org/10.52202/085713-3459

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code is available at https://github.com/RUCAIBox/EASYEP.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We give experiment settings in Section 5.1 and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not provide this.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computer resources in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

https://doi.org/10.52202/085713-3459

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite existing papers and URLs. We also point out their license in Table 8.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We will release the code and documentation after our paper has been accepted.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/085713-3459

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We only employ LLMs for polishing our writing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.