Learning How to Align: Elastic Domain Adaptation for Improving Reasoning Robustness of Large Language Models

Yujie Hou, Mei Wang*, Yaoyao Zhong, Guozhang Li, Qiannan Zhu, Hua Huang
School of Artificial Intelligence, Beijing Normal University
Beijing Key Laboratory of Artificial Intelligence for Education
Engineering Research Center of Intelligent Technology and
Educational Application, Ministry of Education
houyujie@mail.bnu.edu.cn, {wangmei1, zhongyy}@bnu.edu.cn
{liguozhang097, zhuqiannan, huahuang}@bnu.edu.cn

Abstract

Despite exceptional performance on static benchmarks, Large Language Models (LLMs) exhibit significant brittleness under dynamic perturbations. Current extrinsic interventions, *e.g.*, prompt engineering or supervised fine-tuning, overlook the intrinsic cause of this degradation: the distributional shifts in layer activations between original and perturbed questions. To address this, this work formulates the reasoning robustness challenge as an Unsupervised Domain Adaptation (UDA) problem and proposes the Elastic Domain Adaptation (EDA) method. Unlike rigid strategies that enforce a uniform alignment constraint, EDA dynamically aligns layer activations based on the specific layers and perturbation types. Specifically, EDA incorporates an Adaptive Layer Selection (ALS) to identify pivotal layers and employs an Elastic Shift Module (ESM) to selectively pull or push representations in the layer activation space depending on the perturbation types. Extensive experiments across six LLMs and 15 perturbation types demonstrate that EDA achieves state-of-the-art performance on most perturbed benchmarks while preserving reasoning capability on original benchmarks. Code is available at https://github.com/sapphireh/EDA.

1 Introduction

While Large Language Models (LLMs) achieve remarkable performance on static reasoning benchmarks, *e.g.*, mathematical or commonsense tasks, they exhibit marked vulnerability on *dynamic* benchmarks involving perturbations (Zhu et al., 2024; Li et al., 2024b; Huang et al., 2025). This significant performance degradation underscores the intrinsic vulnerability of LLMs when facing the variability of real-world reasoning tasks. To mitigate this, prompt engineering attempts to guide models via refined instructions but often exhibits limited generalizability, showing limited consistency across diverse perturbations (Chatterjee et al., 2024; Li et al., 2023). Similarly, supervised fine-tuning requires costly annotation and computation resources, and may compromise the model's pre-trained reasoning capabilities (Li et al., 2024a). In essence, these methods primarily rely on extrinsic interventions, largely ignoring the intrinsic shifts in feature distributions induced by perturbations.

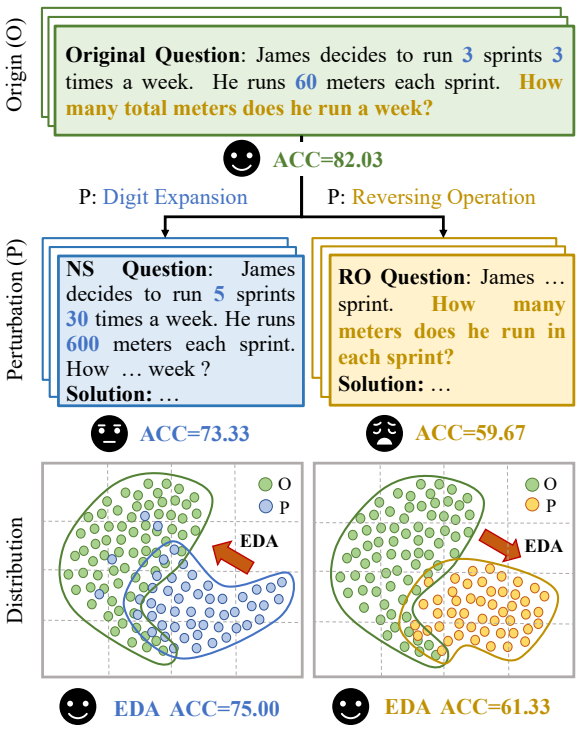


Figure 1: Illustration of distributional shifts induced by perturbations. Perturbations cause the model's activation states to deviate from the original distribution, leading to accuracy degradation. EDA can elastically align these deviations to improve reasoning robustness.

To investigate the intrinsic root cause of this brittleness, we empirically analyze the model's layer activations, which encapsulate the understanding of input sequences (Radford et al., 2019; Gao et al., 2025a). Contrary to the intuition that logic-preserving perturbations (*e.g.*, Digit Expansion)

*Corresponding author

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 8995–9018
October 24-29, 2026 ©2026 Association for Computational Linguistics

should yield consistent representations, Fig. 1 reveals pronounced distributional shifts between the original and perturbed questions. Motivated by this misalignment, we reframe robustness enhancement as an Unsupervised Domain Adaptation (UDA) task. Specifically, original questions are treated as the source domain, while perturbed questions constitute the unlabeled target domain, where perturbations induce systematic distributional shifts. The primary objective of UDA is to induce domain-invariant representations, thereby facilitating the transfer of reasoning capabilities to perturbed data. Notably, this unsupervised paradigm eliminates the reliance on ground-truth labels, allowing us to leverage large-scale perturbed data while circumventing prohibitive annotation costs.

Nevertheless, simply enforcing a rigid alignment between these domains proves suboptimal due to the significant heterogeneity observed in Fig. 2. Our analysis reveals that distributional shifts vary across network depths and perturbation types. Particularly in scenarios where perturbations fundamentally alter the reasoning logic (*e.g.*, Reversing Operating), rigid alignment risks inducing negative transfer. To address this, we introduce Elastic Domain Adaptation (EDA), which employs an elastic strategy to balance the pursuit of domain invariance with the preservation of distinct reasoning paths. Our framework implements elasticity in two aspects. The Adaptive Layer Selection (ALS) introduces elasticity along the model depth by disregarding rigid full-layer alignment to dynamically identify robustness-critical layers. The Elastic Shift Module (ESM) operates within the feature space to address perturbation diversity. It modulates activation distributions by selectively pulling them toward the source or pushing them away—depending on the perturbation characteristics—thereby avoiding the pitfalls of uniform alignment.

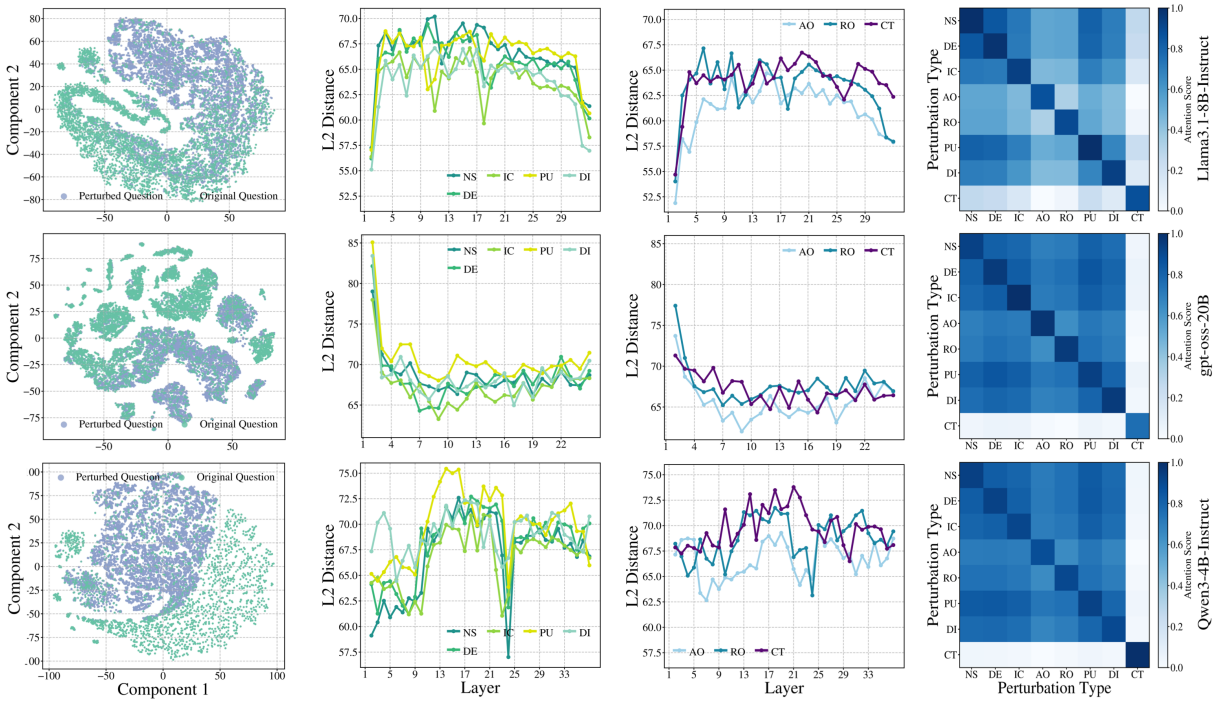
Extensive experiments conducted on GSM8K, MATH, and MMLU across six LLMs and 15 perturbation types demonstrate the effectiveness of our approach. Compared to prompt engineering and fine-tuning methods, EDA achieves state-of-the-art performance on the majority of perturbed tasks, while preserving competitive accuracy on the original benchmarks. Our contributions are:

- We empirically attribute the degradation in LLM reasoning robustness under perturbations to distributional shifts in activations.
- We propose Elastic Domain Adaptation (EDA), a framework that integrates Adaptive Layer Selection to pinpoint critical layers and an Elastic Shift Module to dynamically align distributions based on perturbation types.
- Extensive evaluations across six LLMs and 15 perturbation types demonstrate that EDA achieves state-of-the-art performance on most perturbed benchmarks while preserving performance on original tasks.

2 Related Work

Recent advanced LLMs have achieved remarkable performance on various static evaluation benchmarks (Wang et al., 2024; Suzgun et al., 2023; Srivastava et al., 2023). However, concerns have emerged regarding whether these models possess genuine reasoning capabilities or merely rely on rote memorization of training corpora. To assess true reasoning robustness, dynamic evaluation protocols have been proposed to generate unseen benchmarks. For instance, (Zhu et al., 2024) leveraged dual agents to generate and judge new problems based on psychometric theory, while (Mirzadeh et al., 2025) constructed improved benchmarks using symbolic templates. Additionally, (Huang et al., 2025) employed expert annotators to redesign challenging variants of original problems. Crucially, these studies have collectively demonstrated that LLMs suffer from substantial performance degradation on dynamic benchmarks, highlighting their intrinsic lack of robustness.

Prompt engineering techniques such as Chain-of-Thought (CoT) (Wei et al., 2022) and In-Context Learning (ICL) (Brown et al., 2020; Ma et al., 2025) have been applied to improve model robustness. However, (Zhu et al., 2024) observed that improvements remain limited and inconsistent across different models. Besides, Compositional prompting (COMP) (Li et al., 2024b) attempted to decompose questions into subgoals. But such self-refinement methods often prove ineffective for smaller-scale models due to their limited instruction-following capabilities (Zeng et al., 2024). Alternatively, (Gao et al., 2025b) employed reinforcement learning to foster abstract thinking, but this approach incurs high costs for granular annotations. Critically, these methods primarily rely on extrinsic interventions, overlooking the intrinsic representational shifts that fundamentally drive performance degradation.



(a) Activation distribution (b) Logic-Preserving Distance (c) Logic-Altering Distance (d) Cross-perturb heatmap

Figure 2: Visualization and quantitative analysis on GSM8K and GSM-Plus benchmarks. (a) t-SNE visualization of Layer 13 activation distributions for original versus perturbed inputs. (b–c) Mean paired L2 distances quantifying representational shifts under (b) logic-preserving and (c) logic-altering perturbations. (d) Heatmap of Layer 13 activations across all perturbation types.

3 Empirical Study

We adopt layer activations (Radford et al., 2019; Gao et al., 2025a)—the hidden states of the last token at the final generation step—as a proxy for the model's intrinsic reasoning state, as they effectively encapsulate global semantic information.

Distributional Shift under Perturbations. We begin by visualizing the layer activation of the GSM8K (Cobbe et al., 2021) and the GSM-Plus (Li et al., 2024b) in Fig. 2a. A distinct distributional shift is observed between the original and perturbed samples across all three evaluated models. Although some perturbed questions (see Fig. 2c) share the same underlying reasoning logic as the originals, the perturbations disrupt the model's internal states, causing the reasoning trajectory to drift away from the origin. See more visualization in Append. A.3.

Layer-wise Heterogeneity. To quantify the divergence, we calculate the average L2 distance between original-perturbed pairs across different layers. As shown in Fig. 2b and Fig. 2c, the divergence is not uniform but exhibits significant fluctuations across layers. This layer-wise heterogeneity stems from the distinct functional roles within the Transformer architecture, where different layers are responsible for varying levels of lexical processing and abstract reasoning (Zhao et al., 2025; Zhang et al., 2025). The non-uniform distribution of distances implies that specific layers are more susceptible to input perturbations than others. This insight motivates the need for a mechanism that can adaptively identify and align these critical layers, rather than treating the entire network uniformly.

Perturbation-wise Heterogeneity. Beyond layer-wise variations, we investigate how different perturbation categories impact internal activations by computing the attention similarity heatmap (Fig. 2d). First, we observe significant heterogeneity in the heatmap across different perturbation types. This variance implies that treating all perturbed data as a single, homogeneous target domain is suboptimal. Instead, it necessitates a multi-domain adaptation perspective, where the model adapts to diverse distribution shifts simultaneously. Second, perturbations *e.g.*, Adding Operation (AO), and Reversing Operation (RO), exhibit drastically lower similarity scores compared to other perturbations. This is because AO and RO alter the reasoning path, while Problem Understanding (PU)

merely alters linguistic forms. Consequently, an elastic alignment mechanism is essential to effectively align the distributions of logic-preserving perturbations, while accommodating the necessary representational shifts for logic-altering ones. See Append. A.4 for further visualizations.

4 Methods

Since perturbations induce significant internal distributional shifts, bridging this gap is essential for transferring reasoning capabilities to the target domain. Consequently, we formulate the robustness challenge as an Unsupervised Domain Adaptation (UDA) task (Ganin and Lempitsky, 2015), treating original questions as the source domain and perturbed questions as the unlabeled target domain.

4.1 Preliminary

Let $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{N_s}$ denote the source domain, consisting of N_s labeled examples, where $\mathbf{x}_i^s$ represents the original question and $\mathbf{y}_i^s$ is the corresponding ground-truth solution. The target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{N_t}$ comprises N_t unlabeled examples from the perturbed benchmark. Crucially, we conceptualize $\mathcal{D}_t$ as a union of K distinct perturbation types, *i.e.*, $\mathcal{D}_t = \bigcup_{k=1}^{K} \mathcal{D}_t^k$.

The objective of UDA is to jointly minimize the supervised source domain loss in Eq. 1 and the distributional discrepancy between source and target representations in Eq. 2.

$$\mathcal{L}_{\text{source}} = \mathbb{E}_{(\mathbf{x}_i^s, \mathbf{y}_i^s) \sim \mathcal{D}_s} \left[-\log P_\theta(\mathbf{y}_i^s \mid \mathbf{x}_i^s) \right], \quad (1)$$

$$\begin{aligned} \mathcal{L}_{\text{domain}} = &- \sum_{l=1}^{L} \mathbb{E}_{\mathbf{x}_i^s \sim \mathcal{D}_s} \left[\log(D(\mathbf{A}_l^s)) \right] \\ &- \sum_{l=1}^{L} \mathbb{E}_{\mathbf{x}_j^t \sim \mathcal{D}_t} \left[\log(1 - D(\mathbf{A}_l^t)) \right], \end{aligned} \quad (2)$$

where $\mathbf{A}_l^s$ denotes the layer l activation of $\mathbf{x}^s$, and D is a discriminator trained to distinguish between source $\mathbf{A}_l^s$ and target features $\mathbf{A}_l^t$.

However, the vanilla UDA formulation in Eq. 2 enforces a rigid alignment, treating layers and perturbation types uniformly. This contradicts our empirical findings of layer-wise and perturbation heterogeneity. To address this, we design an Elastic Domain Adaptation (EDA) method with two novel components, ALS and ESM (see Fig. 3). The ALS dynamically identifies critical layers for alignment based on the importance weight w_l of layer l, while the ESM contains a domain-specific discriminator D_k and delta Δ_k for target domain k to align logic-preserving perturbations while maintaining necessary divergence for logic-altering ones.

4.2 Adaptive Layer Selection

To address the layer-wise heterogeneity observed in Sec. 3, ALS functions as a differentiable mechanism that dynamically identifies pivotal layers $\mathcal{I}$ with importance weight w by reconciling the model's intrinsic capability patterns with observed alignment discrepancies. Specifically, we formulate the selection policy based on two complementary factors: a learnable bias φ_l that encodes the model's intrinsic reasoning preference, and an empirical distance d_l that captures the task-specific distributional shift.

Formally, we first quantify the empirical distributional shift d_l to capture the alignment prior,

$$d_l = \|\mathbf{A}_l^s - \mathbf{A}_l^t\|_2, \mathbf{A}_l = \text{MeanPool}(\mathbf{HS}_l), \quad (3)$$

where $\mathbf{HS}_l$ denotes the hidden state of layer l, and $\text{MeanPool}(\cdot)$ represents the averaging operation. Then, we compute the selection logit z_l by combining d_l with a learnable parameter φ_l, and balance them via a hyperparameter λ,

$$z_l = \underbrace{\varphi_l}_{\text{intrinsic}} + \lambda \cdot \underbrace{d_l}_{\text{extrinsic}}. \quad (4)$$

To bridge the gap between discrete selection and continuous optimization, the Gumbel-Softmax relaxation (Jang et al., 2017) is applied on z_l to derive a normalized importance score:

$$w_l = \frac{\exp\left((z_l + g_l)/\tau\right)}{\sum_{i=1}^{L} \exp\left((z_i + g_i)/\tau\right)}, \quad (5)$$

where $g_l \sim \text{Gumbel}(0, 1)$ introduces stochastic noise, and τ is the temperature coefficient.

To achieve hard layer selection, the Straight-Through Estimator (Bengio et al., 2013) is utilized with a threshold σ. The set of pivotal layers is $\mathcal{I} = \{l \mid w_l > \sigma\}$, ensuring that only pivotal layers can participate in the subsequent alignment process. The layer importance w_l modulates the contribution of layer l to the domain loss in Eq. 9.

4.3 Elastic Shift Module

The ESM is designed to achieve fine-grained, elastic alignment tailored to specific perturbation types on the pivotal layers $\mathcal{I}$ from ALS. Standard adversarial adaptation typically employs a single discriminator, assuming a uniform domain shift. However, our analysis reveals significant heterogeneity

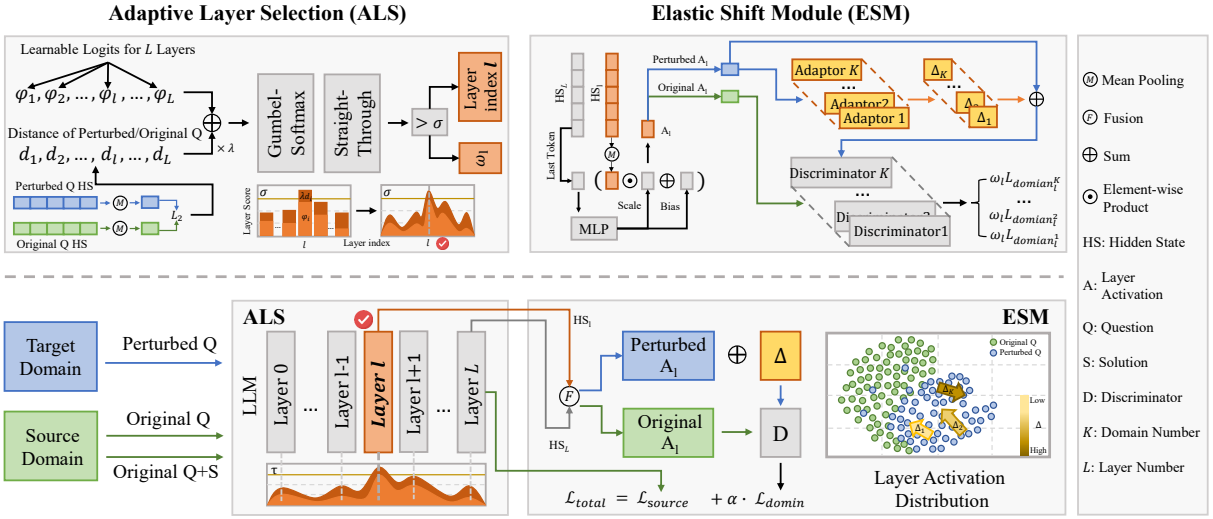


Figure 3: Overview of the Elastic Domain Adaptation (EDA) framework. Key components include: (1) ALS, which selects vital layers via distributional distances; and (2) ESM, which fuses selected layers with the final layer and adds a learnable shift Δ_k to the target domain, achieving elastic alignment via multi-domain adversarial training.

across perturbations. Forcing a single discriminator to align these multimodal distributions leads to optimization interference. To address this, we decompose the target domain into K sub-domains, employing K domain-specific discriminators (Pei et al., 2018). Then, for the heterogeneity between logic-preserving and logic-altering perturbations, we design K adapters to generate a perturbation-specific residual vector Δ_k. Consequently, we align the adapted target $\mathbf{A}_l^{t,k} + \Delta_k$ with the source $\mathbf{A}_l^s$, rather than directly aligning the raw target $\mathbf{A}_l^{t,k}$,

$$\mathbf{A}_l^{t,k} + \Delta_k \xrightarrow{\text{align}} \mathbf{A}_l^s, \tag{6}$$

where $\Delta_k = \text{Adaptor}_k(\mathbf{A}_l{}^{t,k})$ acts as an elastic buffer for target domain k. For logic-preserving perturbations, it is optimized to bridge the domain gap. Conversely, for logic-altering perturbations, the Δ_k accommodates the necessary domain shifts, preventing the negative transfer that would occur under rigid alignment.

Furthermore, motivated by Conditional Domain Adaptation (Long et al., 2018), we identify that aligning the intermediate layers solely is sub-optimal, as these layers may lack the necessary context for the solution. In contrast, the last-layer activation $\mathbf{A}_L$ conveys discriminative information that effectively encapsulates the intent of the solution. Therefore, we utilize the last layer activation $\mathbf{A}_L^{t,k}$ as the solution condition, and fuse the layer activations $\mathbf{A}_l^{t,k}$ selected by ALS with this solution condition via the FiLM (Perez et al., 2018).

$$\hat{\mathbf{A}}_l^{t,k} = \gamma \mathbf{A}_l^{t,k} + \beta + \Delta_k, \tag{7}$$

$$\gamma = f(\mathbf{A}_L^{t,k}), \beta = h(\mathbf{A}_L^{t,k}), \tag{8}$$

where $f(\cdot)$ and $h(\cdot)$ are projection networks that map the condition $\mathbf{A}_L^{t,k}$ to the scale vector γ and shift vector β, respectively. The fused representations $\hat{\mathbf{A}}_L^{t,k}$ are aligned to enable the model to simultaneously capture variations in both the question encoding and the solution logic. The yielding representations $\hat{\mathbf{A}}_l^{t,k}$ to be used in Eq. 9.

4.4 Loss Function

The total loss of EDA is a weighted combination of a supervised source domain loss (Eq. 1) and a multi-domain adversarial domain adaptation loss:

$$\begin{aligned}\mathcal{L}_{\text{domain}} = &-\sum_{l\in\mathcal{I}} w_l \mathbb{E}_{\mathbf{x}_i^s \sim \mathcal{D}_s} \log(D_k(\mathbf{A}_l^s)) \\ &-\sum_{k=1}^{K}\sum_{l\in\mathcal{I}} w_l \mathbb{E}_{\mathbf{x}_j^{t,k} \sim \mathcal{D}_t^k} \log(1 - D_k(\hat{\mathbf{A}}_l^{t,k})),\end{aligned} \tag{9}$$

where A_l^s denotes the layer l activation of $\mathbf{x}^s$, and Δ_k is the elastic shift for perturbation k. D is the discriminator, and w_l denotes the weight to layer l. We employ a Gradient Reversal Layer (Ganin and Lempitsky, 2015) to implement the optimization. The final optimization objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{source}} + \alpha \mathcal{L}_{\text{domain}}, \tag{10}$$

where α is a weight parameter for domain loss.

Methods	GSM8K	GSM-Plus	PRR (%) ↑	HM ↑	MATH	MATH-P	PRR (%) ↑	HM ↑	MMLU	MMLU-D	PRR (%) ↑	HM ↑
Llama3.1-8B-Instruct												
CoT	82.03	64.67	88.16	72.32	**26.43**	15.96	75.30	**19.90**	67.07	64.73	**98.22**	65.87
COMP	73.09	54.42	85.35	62.38	13.04	8.70	**80.00**	10.43	63.79	57.90	95.15	60.70
ICL	/	55.71	/	/	/	**17.28**	/	/	/	54.07	/	/
SFT	79.15	63.75	89.22	70.62	20.43	12.61	76.33	15.59	65.85	63.78	98.40	64.79
EDA	**83.24**	**67.92**	**89.86**	**74.80**	25.87	15.22	74.08	19.16	**67.54**	**64.81**	97.93	**66.14**
gpt-oss-20B												
CoT	95.22	84.38	93.39	89.47	85.65	79.14	96.04	82.27	80.14	78.08	98.69	79.09
COMP	86.20	77.71	94.48	81.73	60.87	55.22	95.13	57.90	64.42	62.62	98.58	63.50
ICL	/	85.54	/	/	/	58.70	/	/	/	78.53	/	/
SFT	94.16	83.12	93.77	88.29	85.96	80.87	96.69	83.33	66.82	62.80	96.89	64.74
EDA	**95.45**	**85.83**	**94.69**	**90.38**	**86.52**	**81.74**	**97.16**	**84.06**	**81.92**	**80.20**	**98.94**	**81.05**
Qwen3-4B-Instruct												
CoT	91.15	78.42	84.30	92.49	81.74	66.52	89.73	73.34	75.35	71.84	97.61	73.55
COMP	89.16	82.04	85.45	**95.84**	56.52	46.52	90.29	51.03	75.46	45.98	75.72	57.14
ICL	/	75.33	/	/	/	56.52	/	/	/	72.61	/	/
SFT	92.19	82.42	87.03	94.40	80.83	70.39	93.09	75.24	73.09	70.75	98.37	71.90
EDA	**92.72**	**83.08**	**87.64**	94.52	**82.17**	**74.35**	**95.00**	**78.06**	**75.55**	**73.74**	**98.79**	**74.63**
Qwen3-4B-Thinking												
CoT	94.09	85.50	95.21	89.58	82.61	70.87	92.35	76.29	77.42	74.95	98.37	76.16
COMP	93.03	85.42	**95.73**	89.06	80.28	73.04	**95.28**	76.48	79.11	74.13	96.75	76.53
ICL	/	85.29	/	/	/	63.91	/	/	/	73.61	/	/
SFT	94.69	83.83	93.39	88.92	80.43	68.26	91.82	73.84	78.98	76.06	98.11	77.49
EDA	**95.30**	**86.29**	95.03	**90.57**	**84.35**	**73.48**	93.11	**78.54**	**79.28**	**76.47**	**98.20**	**77.85**
DeepSeek-R1-Distill-Llama-8B												
CoT	74.07	59.29	88.91	65.86	69.61	53.48	86.89	60.48	67.09	65.30	98.64	66.18
COMP	53.68	45.71	91.98	49.37	64.78	54.78	91.36	59.36	64.58	62.43	98.30	63.48
ICL	/	59.50	/	/	/	49.57	/	/	/	61.97	/	/
SFT	84.23	46.50	71.13	59.92	**70.48**	55.22	87.85	61.92	56.04	55.69	**99.68**	55.86
EDA	**85.97**	**69.08**	**89.11**	**76.61**	68.26	**57.83**	**91.73**	**62.61**	**67.99**	**66.38**	98.80	**67.18**
Qwen2.5-Math-7B-Instruct												
CoT	92.19	79.08	92.34	85.13	87.12	75.28	92.70	80.76	75.45	72.07	97.70	73.72
COMP	90.45	81.29	**94.66**	85.62	64.38	41.30	78.16	50.31	75.42	72.04	97.70	73.69
ICL	/	77.62	/	/	/	49.57	/	/	/	**72.62**	/	/
SFT	92.22	82.88	**94.66**	87.30	73.04	64.35	93.67	68.42	72.91	70.54	**98.34**	71.70
EDA	**92.80**	**83.04**	94.45	**87.65**	**88.26**	**78.35**	**94.05**	**83.01**	**75.49**	72.14	97.73	**73.78**

Table 1: The performance of EDA and baseline methods on the original and perturbed benchmarks. The highest and second-highest accuracies are highlighted in bold and underlined, respectively. Note that ICL and SFT benefit from additional information (source solutions or target solutions) that is unavailable to the unsupervised EDA.

5 Experimental Results

5.1 Experimental Setup

Datasets. We evaluate EDA on six datasets comprising three original benchmarks (GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al.), MMLU (Hendrycks et al., 2021)) and their corresponding perturbed variants (GSM-Plus (Li et al., 2024b), MATH-Perturb (Huang et al., 2025), MMLU-D (Zhu et al., 2024)). This diverse collection covers challenges from math reasoning to general knowledge, enabling a rigorous assessment of model reasoning robustness. The detailed perturbations are introduced in the Append. A.

Baseline Methods. We compare our method with prompt engineering and supervised fine-tuning methods. For prompt engineering, we evaluate Chain-of-Thought (CoT) (Wei et al., 2022), COMP(Li et al., 2024b), and In-Context Learning (ICL). Specifically for ICL, following Huang et al. (2025), we utilize the original problem and solution as a one-shot context. Note that this setting is inapplicable to source benchmarks, where the context would reveal the ground truth. Furthermore, we include Supervised Fine-Tuning (SFT), which is trained explicitly on the perturbed datasets, to serve as a supervised upper-bound baseline.

Evaluation Metrics. We employ four metrics to assess model performance comprehensively. Let ACC_S and ACC_T denote the accuracy on the

source and target benchmarks, respectively.

To evaluate the trade-off and overall balance between source and target domains, we utilize the Harmonic Mean, $HM = \frac{2ACC_S\ ACC_T}{ACC_S+ACC_T}$. Compared to the arithmetic mean, HM is more sensitive to performance disparities, penalizing models that suffer severe degradation on perturbed tasks.

To quantify the robustness relative to the source performance, we measure the Performance Retention Rate, $PRR = \frac{2ACC_T}{ACC_S+ACC_T}$. A higher PRR indicates that the model effectively maintains its capabilities under perturbations.

5.2 Main Results

Tab. 1 presents the comprehensive performance of EDA and baseline methods across six benchmarks. **Reasoning Robustness Improvements.** EDA achieves state-of-the-art ACC_S and ACC_T in the vast majority of settings across diverse model architectures. Remarkably, it frequently surpasses the supervised SFT and ICL methods, despite operating without privileged target labels. This robustness is evident across general instruction-tuned, reasoning-enhanced, distilled, and math-tuned models. Furthermore, EDA achieves the highest PRR and HM in almost all experimental settings, confirming its holistic effectiveness in mitigating distributional shifts while preserving the core reasoning capabilities of the source domain.

Impact of Base Model Capability. On the Llama3.1-8B-Instruct, EDA shows limited improvement on the MATH-P benchmark. We attribute this to the model's inherently low baseline capability on the original unperturbed benchmarks. When the base model struggles to establish a correct reasoning path for the source problem, aligning the target representation to a suboptimal source distribution yields minimal benefit. Conversely, on the Qwen2.5-Math-7B-Instruct, EDA successfully leverages its strong mathematics priors, boosting performance on MATH-P from 75.28% to 78.35%.

5.3 Effect of Different Perturbation Types.

To provide a granular analysis of reasoning robustness, we investigate the impact of specific perturbation types. We evaluate performance across 15 distinct perturbation types derived from GSM-Plus, MATH-Perturb, and MMLU-D benchmarks. The comprehensive comparisons are visualized in Fig. 4 and full results are provided in the Append. A.9.

Robustness against Reasoning-Altering Perturbations. As illustrated in the Fig. 4, EDA achieves the highest accuracy across the vast majority of perturbation types compared to Prompt Engineering and SFT baselines. EDA's superiority is particularly pronounced in perturbations that alter the underlying reasoning logic, such as Reversing Operation, Critical Thinking in GSM-Plus, and Hard Perturbation in MATH-Perturb. For instance, in the Critical Thinking task, EDA achieves an accuracy of 62%, significantly outperforming CoT, which scores only 43%. This substantial gap demonstrates that our proposed elastic alignment strategy effectively enables the model to internalize generalized reasoning patterns and adapt to logic shifts, rather than merely memorizing training templates from the source domain.

Vulnerability of ICL. ICL performs well on Problem Understanding (PU) and Distractor Insertion (DI). As these perturbations preserve the original reasoning path, the provided ICL context acts as a direct shortcut, allowing the model to retrieve the solution without reasoning. However, when the underlying logic is altered, *e.g.*, Reversing Operation (RO), the provided context turns into a misleading prior, causing negative transfer. Consequently, ICL's performance drops drastically, whereas EDA maintains robust reasoning performance.

Method	GSM8K	GSM-Plus
Ablation on EDA		
SFT (Source)	82.70	62.12
SFT (Target)	79.15	63.75
Vanilla UDA	75.22	59.04
UDA + ESM	81.25	66.54
UDA + ESM + ALS (EDA)	**83.24**	**67.92**
Ablation on ESM		
w/o Feature Fusion	82.61	66.58
w/o Adaptor Projection	82.56	67.29
w/o Multi-Adaptor Design	82.63	67.11
w/o Multi-Discriminator	82.79	66.58
w/o Discriminator Pre-training	82.94	67.54
Ablation on ALS		
w/o Distance Scoring	82.98	66.98
w/o Straight-Through Estimator	82.79	67.21

Table 2: Ablation study comparing the full EDA method with SFT baselines and variants in which specific components are removed.

5.4 Ablation Study

To verify the contribution of each component within EDA, we conducted an ablation study using the Llama3.1-8B-Instruct model. The "SFT (Source)" and "SFT (Target)" denote training on the training splits of the source dataset and the target dataset. The results are summarized in Tab. 2.

Effectiveness of the overall framework. Vanilla UDA yields a performance drop compared to the

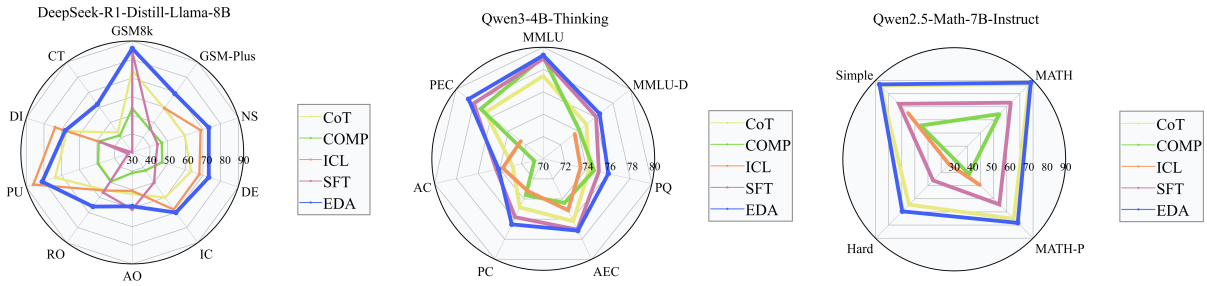


Figure 4: The accuracy of LLM across various original and perturbed questions from GSM-Plus, MATH-Perturb, and MMLU-D. We compare EDA with prompt engineering (CoT, COMP, and ICL) and SFT baselines.

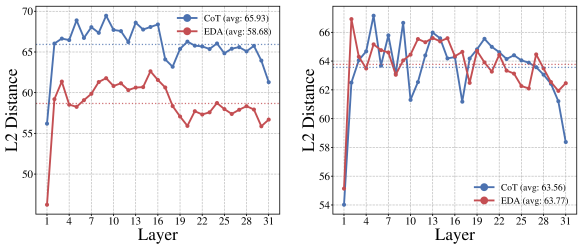
unadapted SFT (Source) baseline. This degradation confirms that rigid alignment induces negative transfer, as it forces the model to align features indiscriminately even when perturbations fundamentally alter the reasoning logic. In contrast, upon integrating ESM and ALS, our EDA achieves a significant improvement (67.92%). Remarkably, EDA surpasses the supervised SFT (Target) baseline (63.75%), while maintaining the highest accuracy on the source domain. This indicates the importance of selectively aligning pivotal layers rather than forcing a rigid full-layer adaptation, and the elastic mechanism successfully reduces the distributional gap without compromising the model's intrinsic reasoning capabilities, offering a superior trade-off between robustness and knowledge retention compared to rigid supervision.

Effectiveness of the ESM and ALS. For ESM, ablating the feature fusion, adaptor, or discriminator pre-training all result in accuracy degradation. Furthermore, removing the Multi-Adaptor or Multi-Discriminator (Multi-D) designs causes distinct performance loss. This confirms that a single rigid projection is insufficient to capture the heterogeneity of perturbation types; instead, a multi-dimensional, elastic alignment strategy is essential for robust adaptation. Regarding ALS, ablating the distance scoring or the Straight-Through Estimator also degrades performance, highlighting the necessity of distribution priors and stable discrete optimization. Additional ablation studies are provided in the Append. A.10.

5.5 Visualization Analysis of EDA

To verify the elasticity of our approach, we quantified activation shifts using the mean paired L2 distance between source and perturbed samples.

As presented in Fig. 5, for the logic-preserving perturbation, Digit Expansion, EDA significantly reduces the representational distance compared to CoT. This indicates that EDA effectively filters out



(a) Digit expansion (b) Reversing operation

Figure 5: Layer-wise distribution distance comparison between EDA and CoT on the GSM-Plus benchmarks.

surface-level noise, pulling the target representation closer to the source manifold. Conversely, for the Reversing Operation, which necessitates a fundamental shift in the reasoning path, EDA exhibits a greater distributional distance than CoT. This indicates that EDA mitigates negative transfer by avoiding over-alignment to a misleading source domain. By preserving necessary divergence, it ensures that the model accommodates the shift in reasoning logic rather than blindly aligning with the source's inapplicable logical shortcut.

6 Conclusion

This paper presented Elastic Domain Adaptation (EDA), a novel framework that formulates reasoning robustness as an unsupervised domain adaptation task. Unlike prior approaches that rely on extrinsic interventions, EDA intrinsically rectifies the distributional discrepancies in model activations. EDA features an elasticity mechanism that selectively identifies robustness-critical layers and elastically modulates the alignment strategy tailored to the heterogeneity of perturbations. This approach effectively mitigates distributional discrepancies while avoiding negative transfer in logic-altering scenarios. Extensive experiments demonstrate the efficacy of EDA for improving LLM reasoning robustness and achieving state-of-the-art performance in most dynamic reasoning benchmarks.

7 Limitations

This study identifies two primary limitations of the proposed EDA framework. First, the method exhibits a dependency on the baseline performance of the LLMs. When the model demonstrates poor capability on the original benchmark, the improvement yielded by EDA is limited. This is because effective domain adaptation requires a knowledgeable baseline to provide reliable signals for target domain alignment. Second, while our approach eliminates the need for ground-truth labels, it needs perturbed input variations for training. This requirement introduces additional computational cost for data synthesis and makes the model's performance partially dependent on the diversity and semantic quality of the generated perturbations.

8 Acknowledgements

This work was supported by the National Natural Science Foundation of China (62437001 and 62402051) and the Fundamental Research Funds for the Central Universities (2253500001).

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Lang Gao, Jiahui Geng, Xiangliang Zhang, Preslav Nakov, and Xiuying Chen. 2025a. Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25378–25398.

Silin Gao, Antoine Bosselut, Samy Bengio, and Emmanuel Abbe. 2025b. Abstral: Augmenting llms' reasoning by reinforcing abstract thinking. *arXiv preprint arXiv:2506.07751*.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, and 1 others. 2025. Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations. In *Forty-second International Conference on Machine Learning*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024a. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308, Miami, Florida, USA. Association for Computational Linguistics.

Moxin Li, Wenjie Wang, Fuli Feng, Yixin Cao, Jizhi Zhang, and Tat-Seng Chua. 2023. Robust prompt optimization for large language models against distribution shifts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1539–1554, Singapore. Association for Computational Linguistics.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024b. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2961–2984.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31.

Xuetao Ma, Wenbin Jiang, and Hua Huang. 2025. Problem-solving logic guided curriculum in-context learning for LLMs complex reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8394–8412, Vienna, Austria. Association for Computational Linguistics.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*.

OpenAI. 2025. Introducing openai gpt-5.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2018. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, and 1 others. 2024. Mr-ben: A meta-reasoning benchmark for evaluating system-2 thinking in llms. *Advances in Neural Information Processing Systems*, 37:119466–119546.

Jue Zhang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025. From reasoning to answer: Empirical, attention-based and mechanistic insights into distilled deepseek r1 models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3985–4002.

Dezhi Zhao, Xin Liu, Xiaocheng Feng, Hui Wang, and Bing Qin. 2025. Probing and boosting large language models capabilities via attention heads. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28518–28532.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024. Dynamic evaluation of large language models by meta probing agents. In *International Conference on Machine Learning*, pages 62599–62617. PMLR.

A Appendix

A.1 Datasets

GSM8K & GSM-Plus. GSM8K is a standard benchmark for grade school mathematical problem-solving, comprising 7,473 training and 1,319 testing samples. GSM-Plus (Fig. 6) contains 10,552 variant questions derived from GSM8K through eight distinct perturbation strategies. The definitions of eight perturbation categories in GSM-PLUS are:

- **Numerical Substitution (NS):** Replaces specific numerical values in the question with different numbers while preserving the original textual context and problem validity.
- **Digit Expansion (DE):** Increases the magnitude or digit count of numerical values within the problem, ensuring the text structure and logical validity remain unchanged.
- **Integer-decimal-fraction Conversion (IC):** Transforms integer values into their decimal or fractional equivalents (and vice versa) without altering the semantic meaning or validity of the question.
- **Adding Operation (AO):** Introduces additional reasoning steps by incorporating extra arithmetic operations (addition, subtraction, multiplication, or division) into the problem logic.
- **Reversing Operation (RO):** Inverts the problem logic by treating the original solution as a known premise and designating a previously known variable as the new target answer. This results in a distinct solution path compared to the original.
- **Problem Understanding (PU):** Rephrases the problem statement using different linguistic structures or vocabulary (lexical and syntactic variations) while maintaining the original underlying logic and solution.
- **Distractor Insertion (DI):** Embeds irrelevant information—specifically context-relevant numerical data or conditions—that has no impact on the derivation of the final answer.
- **Critical Thinking (CT):** Removes essential constraints or necessary conditions from the original problem, rendering it unsolvable due to insufficient information.

MATH & MATH-Perturb. MATH-Perturb dataset focuses on complex reasoning by selecting the most challenging Level-5 problems from the original MATH dataset. It includes 164 training pairs and 115 test pairs (original and perturbed), generated via two perturbation methods. Examples are illustrated in Fig. 7. The definitions of two perturbation categories in MATH-Perturb are:

- **Simple perturbations (Simple):** preserve the core reasoning patterns required to solve the problem.
- **Hard perturbations (Hard):** fundamentally alter the problem formulation, rendering the original solution paths invalid and necessitating adapted reasoning logic.

MMLU & MMLU-D. MMLU assesses comprehensive language understanding across 57 diverse tasks (*e.g.*, STEM, humanities), consisting of 13,985 test samples. The auxiliary training split comprises 99,842 multiple-choice questions aggregated from ARC, MC-TEST, OBQA, and RACE. MMLU-D (Fig. 8) is automatically generated by probing and judging agents to transform original problems based on psychometric theory. The definitions of the five perturbation categories in MMLU-D are:

- **Paraphrasing Questions (PQ):** reformulates the question stem using alternative lexical and syntactic structures while strictly preserving the original semantic meaning and core logic.
- **Paraphrasing Choices (PC):** applies semantic rephrasing to the multiple-choice options. This ensures that the intent and correctness (or incorrectness) of each candidate answer remain unchanged, testing the model's robustness to option phrasing.
- **Permuting Choices (PEC):** shuffles the positional order of the answer options. This perturbation aims to mitigate and evaluate the model's sensitivity to positional bias (*e.g.*, a preference for option 'A').
- **Adding Extra Context (AEC):** injects additional, non-essential information into the question stem. While topically relevant, this context does not aid in solving the problem, serving as noise to test the model's information filtering capabilities.

GSM8K

Original Question:
Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

GSM-Plus

1. Numerical Substitution:
Josh decides to try flipping a house. He buys a house for $90,000 and then puts in $60,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

2. Digit Expansion:
Josh decides to try flipping a house. He buys a house for $800,000 and then puts in $500,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

3. Integer-decimal-fraction Conversion
Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 1.5 times. How much profit did he make?

4. Adding Operation
Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs and $20,000 in furniture. This increased the value of the house by 150%. How much profit did he make?

5. Reversing Operation
Josh decided to try flipping the house. He bought a house for $80,000 and then spent $50,000 on repairs, which increased the value of the house. If he made $70,000 after the house appreciated, what was the percentage appreciation of the house?

6. Problem Understanding
Josh embarks on a house flipping venture. He purchases a property for $80,000 and invests an additional $50,000 for renovations. This action boosts the house's worth by 150%. What is the amount of profit he earned from this venture?

7. Distraction Insertion
Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs for half of a month. This increased the value of the house by 150%. How much profit did he make?

8. Critical Thinking
Josh decides to try flipping a house. He buys a house and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

Figure 6: Examples of GSM8K and GSM-Plus. The perturbed positions are highlighted in blue.

- **Adding a New Choice (AC):** introduces an additional distractor to the option set. This new choice is designed to be contextually plausible but incorrect, requiring specific domain knowledge to be distinguished from the correct answer.

A.2 Training Data Curation

Implementing EDA requires training instances consisting of original questions (with ground-truth solutions) and their perturbed counterparts (which do not require annotation). EDA allows training on perturbed questions without requiring their ground-truth labels. However, GSM-Plus and MMLU-D are released exclusively as testing sets. By leveraging the official prompts and codes, we employed

MATH

Original Question:
If the parabola $y_1 = x^2 + 2x + 7$ and the line $y_2 = 6x + b$ intersect at only one point, what is the value of b?

MATH-Perturb

1. Simple Perturbation:
If the parabola $y_1 = x^2 + 7$ and the line $y_2 = 6x + b$ intersect at only one point, what is the value of b?

2. Hard Perturbation:
If the parabola $y_1 = x^2 + 2x + 7$ and the line $y_2 = bx + 6$ intersect at only one point, what is the value of b?

Figure 7: Examples of MATH and MATH-Perturbation. The perturbed positions are highlighted in blue.

GPT-5 (OpenAI, 2025) to synthesize perturbed training questions only derived from the original GSM8K and MMLU datasets.

In contrast, the Supervised Fine-Tuning (SFT) baseline necessitates ground-truth labels for these perturbed inputs. To address this, we recruited three human annotators to provide solutions for the synthesized GSM-Plus training questions. For the synthesized MMLU-D dataset, since the perturbations are designed to be label-preserving (i.e., they do not alter the correctness of the original answer), we directly inherited the ground-truth labels from the original MMLU dataset without further annotation.

A.3 Visualization of Activation Distribution for All Layers

We utilized t-SNE to visualize the activation distributions across all 32 layers of the Llama3.1-8B-Instruct model. The experiments were conducted on the GSM8K and GSM-Plus benchmarks, with results illustrated in Fig. 9 and Fig. 10. These visualizations clearly demonstrate the distributional shifts between the original and perturbed datasets. Notably, the shape of the distributions tends to stabilize in the deeper layers of the network.

A.4 Cross-Perturb Activation Analysis

To investigate the layer activation across perturbation types, we compute the attention similarity (defined as the average dot product of layer activations) between identical problems subject to different perturbations. Experiments are conducted on the GSM-Plus benchmark using the Llama3.1-8B-Instruct model. The resulting heatmaps are presented in Figures 11 and 12. We observe that Adding Operation (AO), Reversing Operation (RO), and Critical Thinking (CT) exhibit consistently lower similarity scores compared to other pertur-

MMLU

Original Question:
House members are more concerned about their committee assignments than are senators because?
Original Choices
A: House members may serve on only one committee at a time
B: Senate committees have relatively little power to alter legislation
C: Bills in the Senate must pass through all committees before reaching the floor
D: The rules make it extremely difficult for representatives to influence legislation on the House floor

MMLU-D

1. Paraphrasing Question :
Why might representatives in the House be more focused on their committee roles compared to senators?

2. Adding Extra Context:
In the intricate dance of politics within the United States Congress, where the intricate web of interests, regional representation, and policy specialization come to the fore, House members often exhibit a behavior pattern indicative of their legislative priorities and strategic positioning. Given this complex backdrop, why might it be said that House members are more concerned about their committee assignments than are senators?

3. Paraphrasing Choice:
A: Representatives in the House can be members of only one committee at any given time.
B: The influence of Senate committees on the modification of bills is relatively minimal.
C: Every piece of legislation in the Senate is required to be reviewed by each committee before it can be debated on the floor.
D: It is highly challenging for House members to have an impact on legislative proceedings directly on the floor due to strict rules.

4. Adding New Choice:
A: House members may serve on only one committee at a time
B: Senate committees have relatively little power to alter legislation
C: Bills in the Senate must pass through all committees before reaching the floor
D: The rules make it extremely difficult for representatives to influence legislation on the House floor
E: House members are required to consult with their constituents before voting in committees

5. Permuting Choice:
A: Bills in the Senate must pass through all committees before reaching the floor
B: The rules make it extremely difficult for representatives to influence legislation on the House floor
C: Senate committees have relatively little power to alter legislation
D: House members may serve on only one committee at a time",

Figure 8: Examples of MMLU and MMLU-D. The perturbed positions are highlighted in blue.

bations across all layers. This divergence suggests that these specific perturbations fundamentally alter the underlying reasoning logic, whereas others primarily effectuate surface-level linguistic changes.

A.5 Cross-Layer Activation Analysis

We compute the cross-layer attention similarity heatmaps to visualize the correlation between activations at different depths for perturbed questions. Experiments are conducted on the GSM-Plus benchmark using Llama3.1-8B-Instruct, with results for all perturbation types illustrated in Fig. 13. We identify a distinct high-similarity block in the deeper layers across all eight perturbation categories, suggesting that the model successfully stabilizes its reasoning path and achieves representation convergence in the final stages. This pattern suggests that regardless of the input perturbation, the model successfully stabilizes its reasoning path and achieves representation convergence in the final stages.

A.6 Architecture of Adapters

To achieve fine-grained feature alignment for different domains, we introduce the Adapter, a deep residual module designed to learn domain-specific feature offsets.

As shown in our implementation, the adapter $\mathcal{A}$ is an MLP-based residual network. Given an input feature vector $\mathbf{f} \in \mathbb{R}^d$, the adapter projects it to a hidden dimension d_h, processes it through L stacked Residual Blocks, and projects it back to the original dimension to generate the offset $\Delta\mathbf{f}$. To ensure training stability and preserve the pre-trained model's original capabilities at the initial phase, we apply Zero Initialization to the weights and biases of the final output projection layer. This ensures the adapter acts as an identity mapping (i.e., $\Delta\mathbf{f} = \mathbf{0}$) at initialization.

To handle multi-domain adaptation, we utilize a multi-adaptor module. It maintains a set of independent adapters $\{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$, where K is the number of domains. During the forward pass, the specific adapter $\mathcal{A}_k$ is activated according to the target domain index k to compute the domain-specific residual.

A.7 Evaluated LLMs

To demonstrate the broad applicability of our method, we apply EDA to six open-source LLMs, including general instruction-tuned models (Llama3.1-8B-Instruct (Dubey et al., 2024), gpt-oss-20b (Agarwal et al., 2025), Qwen3-4B-Instruct (Yang et al., 2025)), reasoning-enhanced models (Qwen3-4B-Thinking, DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025)), and domain-specific experts (Qwen2.5-Math-7B-Instruct (Yang et al., 2024)).

A.8 Implementation Details

We set the domain loss weight α to 0.2, and the distance weight λ to 1 in the ALS. For the Gumbel-Softmax operation, the initial temperature is set to 5.0 and decays to 0.5 via an exponential annealing schedule. The threshold σ for the Straight-Through Estimator is set to 0.1. Comprehensive ablation studies regarding these hyperparameter selections are discussed in the Append. A.10.3.

Our objective is to jointly minimize the source supervision loss and the distributional discrep-

ancy between source and target representations in a multi-adversarial domain adaptation manner. To ensure training efficiency, we adopt QLoRA (Dettmers et al., 2023) to fine-tune the LLM. Additionally, we initialize the discriminator via pre-training to facilitate stable adversarial convergence.

All experiments are conducted on a computational node equipped with two NVIDIA H800 GPUs. For the training configuration, the EDA framework is trained for 2 epochs with a learning rate of 1×10^{-6}. To ensure stable initialization, the discriminator undergoes a pre-training phase of 1 epoch before the joint training.

A.9 Experiments on Fine-grained Perturbations

This section presents a comprehensive performance comparison between our proposed EDA framework and baseline methods (CoT, COMP, ICL, and SFT) across five different LLMs on the fine-grained perturbation benchmarks.

GSM-Plus. The fine-grained results in Tab. 3 reveal that while baseline methods like ICL may excel at specific perturbed questions, they lack the depth to handle fundamental logic-shift questions. EDA stands out by bridging this gap, providing comprehensive reasoning robustness that covers both simple logic-preserving perturbations and complex logic-altering perturbations. For instance, in the CT perturbation, ICL yields near-zero accuracy on Llama3.1-8B-Instruct and Qwen2.5-Math-7B-Instruct, and SFT performance on DeepSeek-R1-Distill-Llama-8B collapses to 7.3%. In contrast, EDA achieves significant gains, indicating it learns intrinsic logic verification rather than superficial pattern matching. Additionally, EDA consistently leads in RO, evidencing enhanced bidirectional reasoning capabilities. On PU and DI perturbations, EDA maintains a balanced robustness profile compared to baselines. Furthermore, compared to SFT, EDA shows consistent improvements on numerical variations (NS, DE).

MATH-Perturb. Tab. 4 compares EDA with baselines on the challenging MATH and MATH-Perturb benchmarks. EDA demonstrates superior performance against logical restructuring perturbations. While SFT suffers significant performance drops on the "Hard" subset (*e.g.*,, collapsing to 46.09% on Qwen2.5-Math-7B-Instruct), EDA maintains high accuracy (69.57%), delivering a substantial 23.48% improvement. Similarly, on DeepSeek-R1-Distill-Llama-8B, EDA outperforms SFT by over 15% on the Hard perturbation benchmark. This confirms EDA's ability to learn transferable reasoning logic beyond mere template memorization. On Simple perturbations and the original MATH dataset, EDA consistently outperforms SFT. Notably, on Qwen2.5-Math-7B-Instruct, EDA surpasses SFT by over 15% on the original benchmark (88.26% vs. 73.04%), proving that evolutionary adaptation enhances both robustness and fundamental mathematical capability without trade-offs.

MMLU-D. The fine-grained breakdown results in Tab. 5 highlight EDA's specific strengths in handling diverse types of noise and structural changes. PEC perturbation tests whether models rely on the position of the answer (*e.g.*,, favoring option "A") rather than the content. EDA achieves the highest accuracy on PEC across all models (*e.g.*, 78.58% on Qwen3-4B-Thinking and 75.26% on Qwen2.5-Math). This suggests that the EDA effectively debiases the model's position preference. For AEC, which introduces irrelevant information, EDA consistently outperforms baselines. On Llama3.1-8B, EDA reaches 66.23%, surpassing CoT (66.13%) and SFT (65.98%). This indicates that EDA improves the model's information filtering capabilities, allowing it to focus on core logic despite noisy inputs.

A.10 Ablation Study

A.10.1 Components in ALS

Ablation on Activation Aggregation Strategy. In this section, we evaluate the impact of different aggregation strategies for computing layer activations within the ALS module. We compare using the hidden state of the last token against applying mean pooling across the entire sequence. As shown in Tab. 6, mean pooling consistently yields superior performance on both GSM8K and GSM-Plus benchmarks. This suggests that a holistic representation (averaged across tokens) captures the semantic features required for domain adaptation more effectively than the representation of a single final token. Consequently, we adopt mean pooling as the default configuration for our framework.

Choice of Distance Metrics. In the ALS module, we employ the L_2 norm to quantify the divergence between original and perturbed layer activations. To validate this choice, we compared L_2 against four alternative metrics: L_1 norm (Manhattan distance), Cosine distance ($1 -$ cosine similarity), Maximum Mean Discrepancy (MMD) (Gretton

Methods	GSM8K	GSM-Plus								
		AVG	NS	DE	IC	AO	RO	PU	DI	CT
Llama3.1-8B-Instruct										
CoT	82.03	64.67	72.33	73.33	67.00	59.67	59.67	81.33	66.67	36.67
COMP	73.09	54.42	56.33	59.67	53.67	48.00	45.67	70.00	55.00	31.00
ICL	/	55.71	61.67	52.33	53.33	55.00	47.33	**95.00**	**76.33**	4.67
SFT	79.15	63.75	67.67	70.33	64.33	57.33	55.33	78.00	65.67	**51.33**
EDA	**83.24**	**67.92**	**73.00**	**75.00**	**72.00**	**61.33**	**61.33**	82.33	72.33	46.67
gpt-oss-20B										
CoT	95.22	84.38	85.33	89.00	88.67	78.33	84.33	94.00	**90.00**	65.33
COMP	86.20	77.71	79.33	83.00	85.33	75.33	81.33	88.00	87.33	42.00
ICL	/	85.54	85.33	87.67	89.00	79.00	**85.67**	**94.33**	89.00	**74.33**
SFT	94.16	83.12	82.67	89.00	89.00	79.33	82.00	92.00	88.00	63.00
EDA	**95.45**	**85.83**	**86.00**	**90.00**	**90.00**	**82.00**	**85.67**	93.00	**90.00**	70.00
Qwen3-4B-Instruct										
CoT	91.15	78.42	80.33	84.67	83.00	72.00	73.00	88.67	83.67	62.00
COMP	89.16	82.04	82.00	**87.67**	85.67	73.33	82.33	89.00	90.00	66.33
ICL	/	75.33	81.33	86.67	84.00	60.67	56.00	**100.00**	**92.67**	41.33
SFT	92.19	82.42	**82.67**	86.67	**87.67**	**75.67**	**83.33**	90.67	88.00	64.67
EDA	**92.72**	**83.08**	81.67	87.00	86.67	74.67	81.67	91.33	88.67	**73.00**
Qwen3-4B-Thinking										
CoT	94.09	85.50	86.33	87.33	**91.33**	78.67	86.33	93.67	90.67	69.67
COMP	93.03	85.42	85.67	88.33	89.67	79.67	87.00	91.33	91.00	**70.67**
ICL	/	85.29	**88.00**	**90.67**	89.67	**81.33**	88.33	**96.67**	92.00	55.67
SFT	94.69	83.83	85.00	89.33	91.00	**81.33**	**88.67**	94.00	82.00	49.33
EDA	**95.30**	**86.29**	87.33	**90.67**	**91.33**	78.67	87.00	94.33	**94.67**	66.33
DeepSeek-R1-Distill-Llama-8B										
CoT	74.07	59.29	59.67	63.33	60.00	52.33	55.00	73.67	67.00	43.33
COMP	53.68	45.71	46.67	47.00	42.00	41.00	49.33	49.33	49.00	41.33
ICL	/	59.50	68.67	68.00	68.00	50.67	55.33	**86.00**	**73.67**	5.67
SFT	84.23	46.50	44.33	43.33	50.00	**60.67**	56.67	32.33	48.00	7.33
EDA	**85.97**	**69.08**	**73.33**	**73.33**	**70.00**	59.00	**66.00**	81.00	68.00	**62.00**
Qwen2.5-Math-7B-Instruct										
CoT	92.19	79.08	81.33	85.00	84.67	72.00	74.67	84.67	84.67	65.67
COMP	90.45	81.29	79.33	84.33	84.33	73.33	82.00	89.33	86.33	71.33
ICL	/	77.62	**89.67**	**90.33**	**91.00**	**78.33**	**85.33**	**98.33**	83.33	4.67
SFT	92.22	82.88	84.67	87.00	88.33	75.67	83.00	90.67	89.00	64.67
EDA	**92.80**	**83.04**	80.33	88.00	86.00	71.67	83.00	92.00	**89.33**	**74.00**

Table 3: Performance comparison on GSM8K and GSM-Plus benchmarks with fine-grained perturbation breakdown. The highest and second-highest accuracies are highlighted in bold and underlined, respectively. Perturbation types include: NS (Numerical Substitution), DE (Digit Expansion), IC (Integer-decimal-fraction Conversion), AO (Adding Operation), RO (Reversing Operation), PU (Problem Understanding), DI (Distractor Insertion), and CT (Critical Thinking).

Methods	MATH	MATH-P		
		AVG	Hard	Simple
Llama3.1-8B-Instruct				
CoT	**26.43**	15.96	**9.43**	22.48
COMP	13.04	8.70	4.35	13.04
ICL	/	**17.28**	8.75	**25.82**
SFT	20.43	12.61	6.09	19.13
EDA	25.87	15.22	6.09	24.35
gpt-oss-20B				
CoT	85.65	79.14	65.57	88.70
COMP	60.87	55.22	51.74	58.70
ICL	/	58.70	53.56	71.26
SFT	85.96	80.87	**73.91**	87.83
EDA	**86.52**	**81.74**	71.57	**90.70**
Qwen3-4B-Instruct				
CoT	81.74	66.52	61.74	71.30
COMP	56.52	46.52	43.04	60.00
ICL	/	56.52	44.35	68.70
SFT	80.83	70.39	61.70	79.09
EDA	**82.17**	**74.35**	**65.22**	**83.48**
Qwen3-4B-Thinking				
CoT	82.61	70.87	60.00	81.74
COMP	80.28	73.04	**61.74**	84.35
ICL	/	63.91	53.04	74.78
SFT	80.43	68.26	58.26	78.26
EDA	**84.35**	**73.48**	60.87	**86.09**
DeepSeek-R1-Distill-Llama-8B				
CoT	69.61	53.48	44.35	62.61
COMP	64.78	54.78	47.78	61.74
ICL	/	49.57	34.78	64.35
SFT	**70.48**	55.22	33.48	56.09
EDA	68.26	**57.83**	**48.70**	**66.96**
Qwen2.5-Math-7B-Instruct				
CoT	87.12	75.28	64.47	86.09
COMP	64.38	41.30	26.96	55.65
ICL	/	49.57	33.91	65.22
SFT	73.04	64.35	46.09	72.17
EDA	**88.26**	**78.35**	**69.57**	**86.95**

Table 4: Performance comparison on MATH and MATH-Perturb benchmarks with fine-grained perturbation breakdown. The highest and second-highest accuracies are highlighted in bold and underlined, respectively. Hard denotes the hard perturbation, and Simple denotes the simple perturbation.

et al., 2012), and CORAL distance (Correlation Alignment) (Sun and Saenko, 2016). The results, detailed in Tab. 7, demonstrate that L_2 achieves the highest performance. This suggests that strictly aligning both the magnitude and direction of activation vectors via Euclidean distance provides the most effective supervision for this task compared to statistical (MMD, CORAL) or direction-only (Cosine) metrics.

Layer Selection In Tab. 10, we compare EDA against various single-layer alignment baselines, ranging from shallow to deep layers. The results demonstrate that EDA consistently outperforms all fixed-layer comparators, validating the effectiveness of our adaptive strategy.

A.10.2 Components in ESM

Ablation on Fusion Mechanisms. In the ESM module, we employ FILM (Perez et al., 2018) to fuse the selected layer activation with the final layer activation. To validate this design choice, we compared FILM against three alternative fusion strategies:

- **Concatenation:** Directly concatenates the two feature vectors, followed by a linear projection and LayerNorm.
- **Cross-Attention:** Utilizes the final layer activation as the query to attend to the selected layer activation.
- **Multilinear Map:** Adopts the randomized multilinear map proposed in CDAN (Long et al., 2018) to capture high-order feature interactions.

The results, presented in Tab. 8, demonstrate that FILM achieves superior performance on both GSM8K and GSM-Plus. We hypothesize that FILM is particularly effective because it applies feature-wise affine transformations (scaling and shifting), which efficiently modulate the distribution of shallow features conditioned on the deep features. In contrast, Concatenation lacks interaction depth, while Multilinear maps and Cross-Attention introduce additional complexity that may lead to optimization difficulties.

A.10.3 Hyperparameter

We performed a systematic ablation study to determine the optimal values for three critical hyperparameters in the EDA framework: the STE threshold σ, the distance weight λ, and the temperature τ. All

Methods	MMLU	MMLU-D					
		AVG	PQ	AEC	PC	AC	PEC
Llama3.1-8B-Instruct							
CoT	67.07	64.73	**63.90**	66.13	63.35	63.68	66.61
COMP	63.79	57.90	56.15	61.02	58.23	54.94	59.16
ICL	/	54.07	56.20	52.50	55.74	52.89	53.01
SFT	65.85	63.78	62.57	65.98	61.68	**64.69**	63.96
EDA	**67.54**	**64.81**	63.70	**66.23**	**63.71**	63.61	**66.77**
gpt-oss-20B							
CoT	80.14	78.08	77.02	79.75	77.43	76.70	79.48
COMP	64.42	62.62	62.02	65.09	61.59	60.44	63.59
ICL	/	78.53	**79.29**	80.44	**79.77**	**80.06**	73.29
SFT	66.82	62.80	61.24	67.12	60.95	61.49	63.18
EDA	**81.92**	**80.20**	78.73	**82.07**	79.56	78.95	**81.72**
Qwen3-4B-Instruct							
CoT	75.35	71.84	71.06	73.62	70.46	70.39	73.67
COMP	75.46	45.98	46.43	44.42	45.28	47.28	46.49
ICL	/	72.61	72.44	74.12	72.21	**72.72**	71.56
SFT	73.09	70.75	69.67	71.43	69.63	70.50	72.54
EDA	**75.55**	**73.74**	**73.06**	**75.63**	**72.55**	71.65	**75.81**
Qwen3-4B-Thinking							
CoT	77.42	74.95	74.24	76.21	74.83	72.83	76.65
COMP	79.11	74.13	74.68	74.39	73.61	70.82	77.18
ICL	/	73.61	73.44	75.12	73.21	73.73	72.56
SFT	78.98	76.06	75.12	77.05	75.81	**74.33**	**77.98**
EDA	**79.28**	**76.47**	**75.99**	**77.15**	**76.52**	74.11	**78.58**
DeepSeek-R1-Distill-Llama-8B							
CoT	67.09	65.30	65.01	67.32	64.71	62.60	66.84
COMP	64.58	62.43	62.22	64.56	61.41	60.44	57.98
ICL	/	61.97	63.72	64.20	63.53	60.44	57.98
SFT	56.04	55.69	55.85	59.41	52.91	54.99	55.29
EDA	**67.99**	**66.38**	**66.19**	**68.55**	**65.51**	**63.91**	**67.77**
Qwen2.5-Math-7B-Instruct							
CoT	75.45	72.07	71.35	73.64	70.81	70.52	74.02
COMP	75.42	72.04	71.95	73.04	70.97	69.57	74.15
ICL	/	**72.62**	**72.59**	**74.09**	**72.20**	**72.83**	71.41
SFT	72.91	70.54	69.42	71.18	69.60	70.23	72.28
EDA	**75.49**	72.14	71.28	72.03	71.75	70.39	**75.26**

Table 5: Performance comparison on MMLU and MMLU-D benchmarks with fine-grained perturbation breakdown. The highest and second-highest accuracies are highlighted in bold and underlined, respectively. Perturbation types include: PQ (Paraphrasing Questions), AEC (Adding Extra Context), PC (Paraphrasing Choices), AC (Adding a New Choice), PEC (Permuting Choice).

Method	GSM8K	GSM-Plus
Last token	82.49	67.04
Mean	**83.24**	**67.92**

Table 6: Ablation Study for layer activation computing methods in ALS. We conduct experiments on the GSM8K and GSM-Plus benchmarks with the Llama3.1-8B-Instruct model.

Distance Metric	GSM8K	GSM-Plus
L1	83.09	66.58
Cosine	82.71	66.92
MMD	82.56	66.46
CORAL	82.34	66.79
L2	**83.24**	**67.92**

Table 7: Ablation Study for distribution distance methods in ALS. Experiments are conducted using the Llama3.1-8B-Instruct model on the GSM8K and GSM-Plus benchmarks.

Fusion Strategy	GSM8K	GSM-Plus
Concatenation	82.94	67.38
Cross-Attention	82.64	67.06
Multilinear Map	82.18	67.67
FiLM	**83.24**	**67.92**

Table 8: Ablation Study for activation fusion methods in ESM. Experiments are conducted using the Llama3.1-8B-Instruct model on the GSM8K and GSM-Plus benchmarks.

experiments were conducted using Llama3.1-8B-Instruct. (see Tab. 9).

- **STE Threshold σ:** Performance peaks at $\sigma = 0.1$. Both 0.3 and 0.05 thresholds degrade accuracy, indicating 0.1 is the optimal balance for layer selection.
- **Distance Weight λ:** Comparing λ values from 0.1 to 1.5, we find that $\lambda = 1$ achieves the best accuracy on both GSM8K and GSM-Plus, effectively balancing feature alignment with the supervised loss.
- **Temperature τ:** We find that an annealing schedule starting at 5 and decaying to 0.5 yields superior results. This gradual sharpening of the probability distribution enables the model to explore diverse layer combinations early in training before converging to a stable selection.

Method	GSM8K	GSM-Plus
STE threshold σ		
0.3	82.71	66.92
0.1	**83.24**	**67.92**
0.05	83.09	66.62
Distance Weight λ		
0.1	83.12	67.29
0.3	82.56	66.62
0.5	82.34	66.83
0.7	82.98	67.29
1	**83.24**	**67.92**
1.2	82.87	67.00
1.5	82.81	67.04
Temperature τ		
inital 5, decay to 1	82.98	66.98
inital 5, decay to 1.5	82.79	67.21
inital 5, decay to 2	83.12	67.33
inital 2, decay to 0.5	82.89	66.67
inital 3, decay to 0.5	83.08	67.00
inital 4, decay to 0.5	82.56	67.12
inital 5, decay to 0.5	**83.24**	**67.92**

Table 9: The ablation study for hyperparameter STE threshold σ, Distance Weight λ and Temperature τ. Experiments are conducted using the Llama3.1-8B-Instruct model on the GSM8K and GSM-Plus benchmarks.

Method	GSM8K	GSM-Plus
Layer 1	82.87	67.00
Layer 4	82.71	67.33
Layer 6	83.14	67.21
Layer 8	82.64	67.12
Layer 10	82.34	67.17
Layer 12	82.56	67.84
Layer 14	83.02	67.12
Layer 16	82.87	66.88
Layer 18	83.02	67.15
Layer 20	82.87	67.42
Layer 22	82.41	66.62
Layer 24	82.49	67.25
Layer 26	82.87	67.25
Layer 28	82.94	67.17
Layer 30	**83.24**	67.46
Layer 32	82.11	66.54
EDA	**83.24**	**67.92**

Table 10: The ablation study for layer selection. Experiments are conducted using the Llama3.1-8B-Instruct model on the GSM8K and GSM-Plus benchmarks.

Intervention	Accuracy (%)	Drop
No activation intervention	82.03	–
Random shift	80.14	1.89
Source-to-target shift	76.58	5.45
Additional degradation	–	**3.56**

Table 11: Effect of activation-shift interventions on reasoning accuracy.

A.11 Interventional Analysis of Activation Shifts

Our empirical analyses reveal a strong and consistent association between perturbation-induced activation distribution shifts and reasoning robustness degradation. We interpret this relationship as a representational account of robustness degradation rather than a complete causal explanation, since reasoning failures may arise from multiple factors. Nevertheless, the observed association motivates EDA to mitigate harmful source–target representational discrepancies and facilitate the transfer of existing reasoning capability to perturbed inputs.

To further examine the functional relevance of activation shifts, we conduct an intervention experiment by injecting the empirically estimated source-to-target shift into the activations of original questions at ALS-selected layers. As shown in Tab. 11, the intervention reduces accuracy from 82.03% to 76.58%, corresponding to a 5.45-point drop. In comparison, a random shift reduces accuracy to 80.14%, a 1.89-point drop. Thus, the perturbation-aligned shift induces an additional 3.56-point degradation beyond generic activation corruption.

A.12 Robustness to Perturbation-Type Granularity

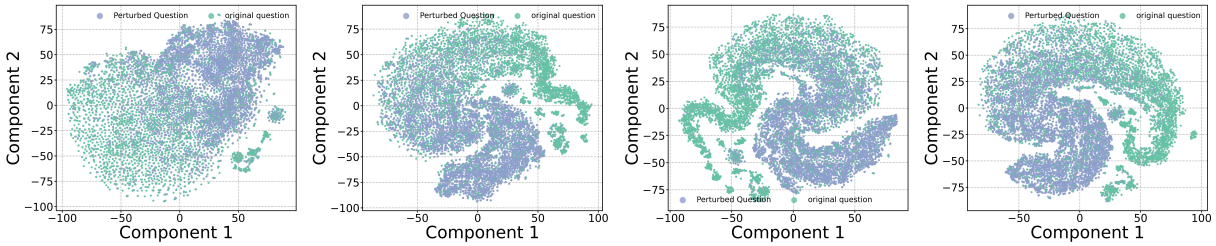
Full EDA partitions the target domain into (K) perturbation-specific subdomains during adaptation. To examine its dependence on fine-grained perturbation annotations, we evaluate EDA under different target-domain partition granularities in Tab. 12. Specifically, (K=1) treats all perturbations as a single mixed target domain, (K=2) groups them into logic-preserving and logic-altering perturbations, and (K=8) follows the eight fine-grained perturbation categories in GSM-Plus.

Even without perturbation-type information, EDA ((K=1)) improves GSM-Plus accuracy by 5.61 points over vanilla UDA while substantially preserving source-domain performance. Using only the coarse logic-preserving/logic-altering partition further improves the target accuracy to 65.54. Fine-grained perturbation information provides additional gains, but the strong performance of (K=1) and (K=2) shows that EDA remains effective under coarse or unavailable perturbation-type information.

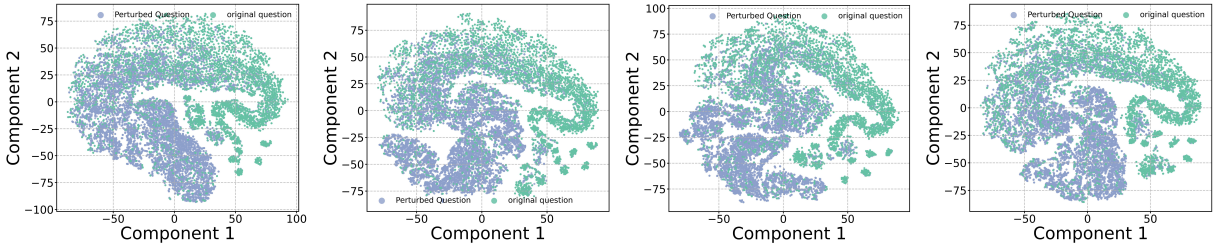
Perturbation types are used only during adaptation to route target samples to the corresponding auxiliary modules. After adaptation, these modules are removed, and inference is performed with the adapted backbone alone; therefore, no perturbation-type information is required at deployment.

Method	Type Information	GSM8K	GSM-Plus
Vanilla UDA	None	75.22	59.04
EDA ($K = 1$)	All perturbations grouped together	81.17	64.65
EDA ($K = 2$)	Logic-preserving / logic-altering	81.65	65.54
EDA ($K = 8$)	Eight fine-grained perturbation types	**83.24**	**67.92**

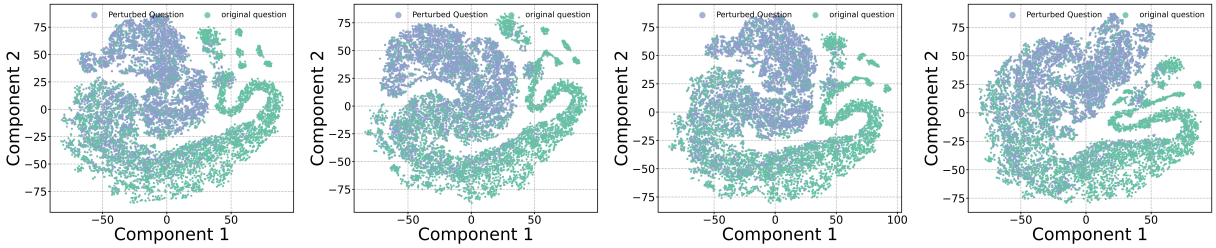
Table 12: Performance of EDA under different perturbation-type granularities on GSM8K and GSM-Plus.



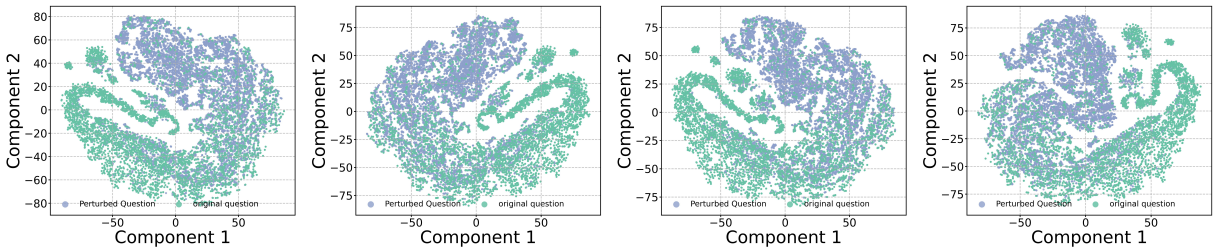
(a) layer 1 to layer 4



(b) layer 5 to layer 8

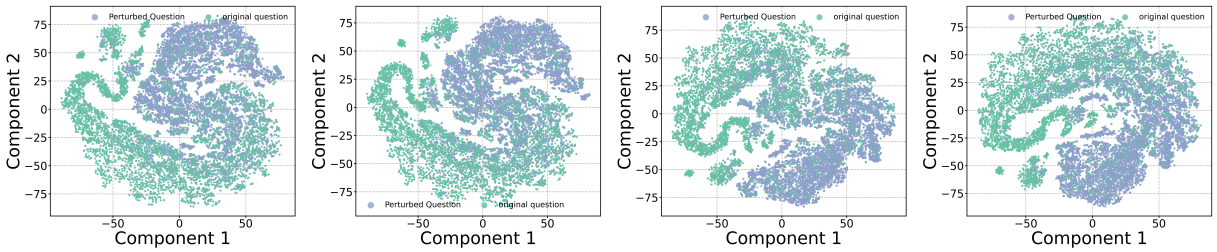


(c) layer 9 to layer 12

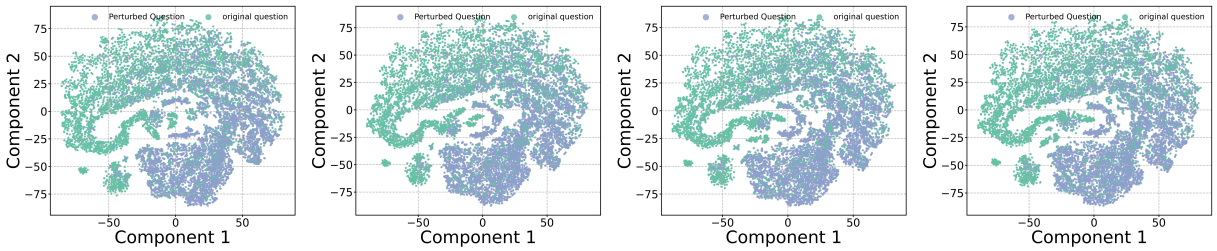


(d) layer 12 to layer 16

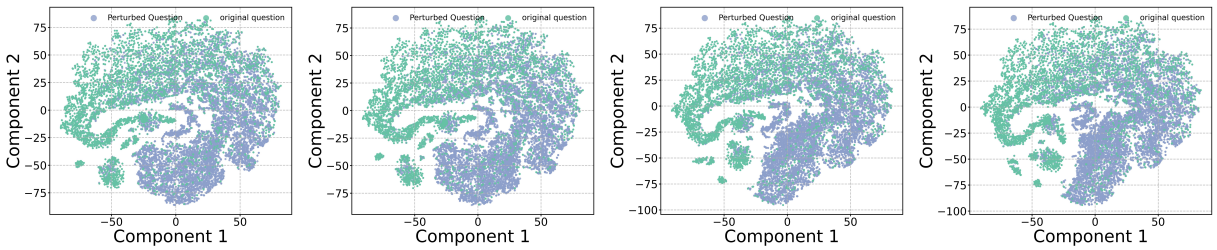
Figure 9: Layer activation distributions for layer 1 to layer 16. Green dots represent the original questions, and blue dots represent perturbed questions. We conduct experiments on the GSM8K and GSM-Plus benchmarks with the Llama3.1-8B-Instruct model.



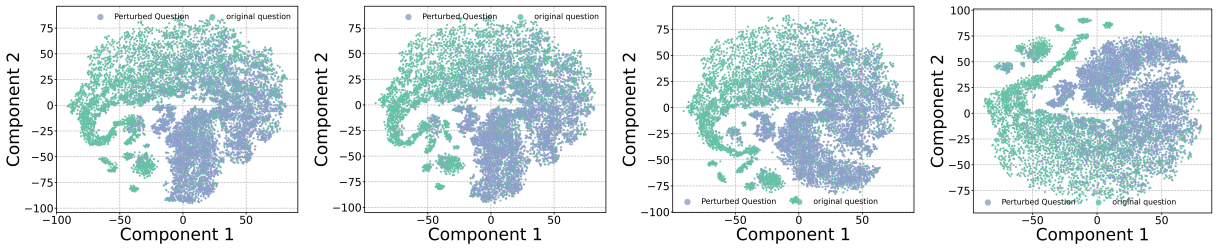
(a) layer 17 to layer 20



(b) layer 21 to layer 24

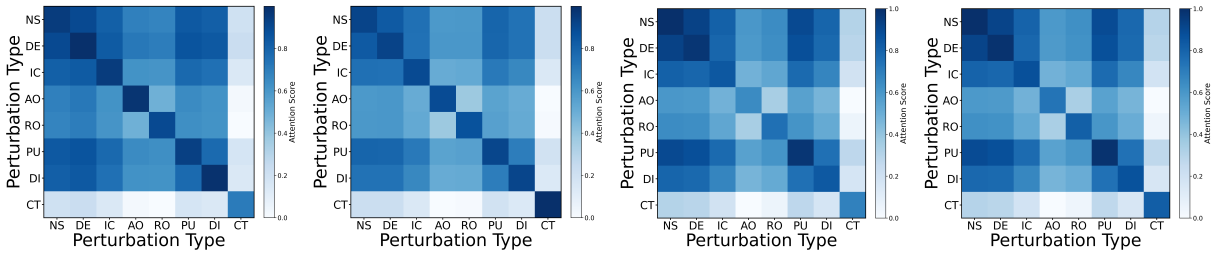


(c) layer 25 to layer 28

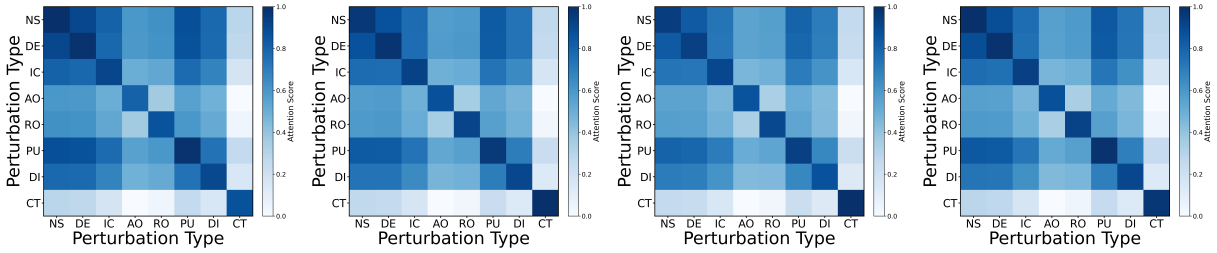


(d) layer 29 to layer 32

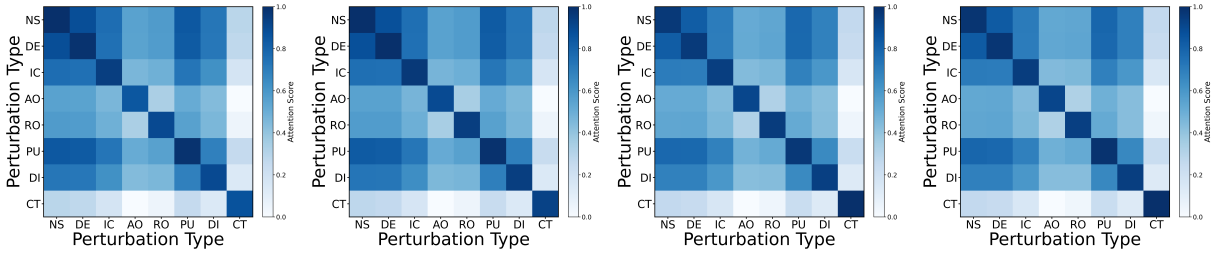
Figure 10: Layer activation distributions for layer 17 to layer 32. Green dots represent the original questions, and blue dots represent perturbed questions. We conduct experiments on the GSM8K and GSM-Plus benchmarks with the Llama3.1-8B-Instruct model.



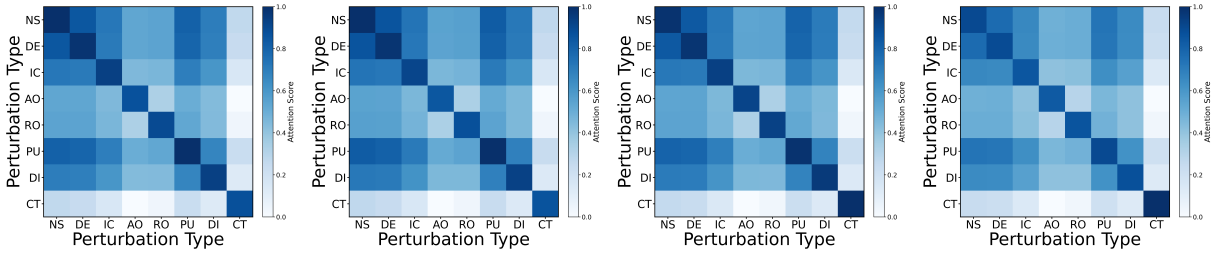
(a) layer 1 to layer 4



(b) layer 5 to layer 8

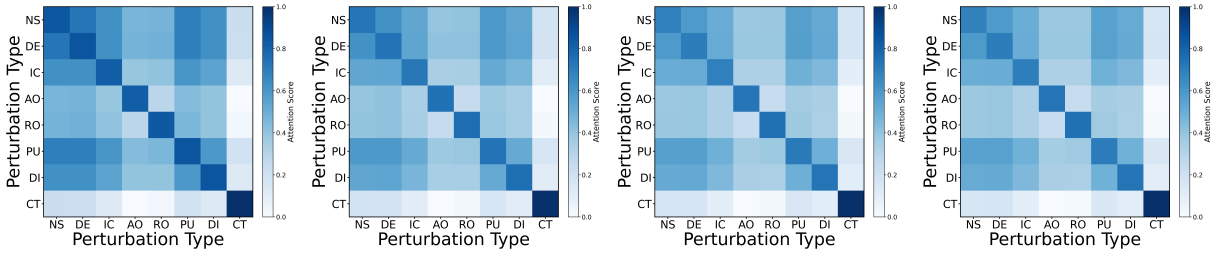


(c) layer 9 to layer 12

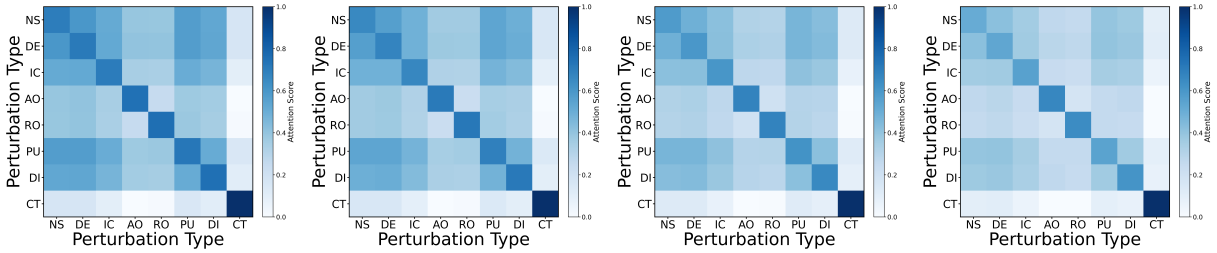


(d) layer 12 to layer 16

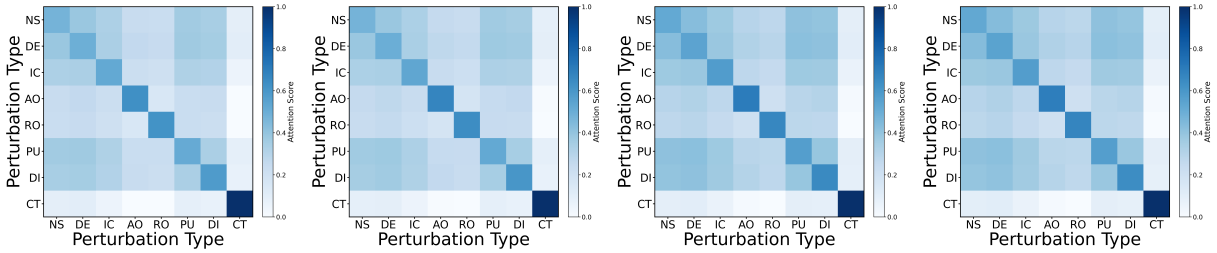
Figure 11: Cross-perturb activation heatmap for layer 1 to layer 16. We conduct experiments on the GSM-Plus benchmark with the Llama3.1-8B-Instruct model.



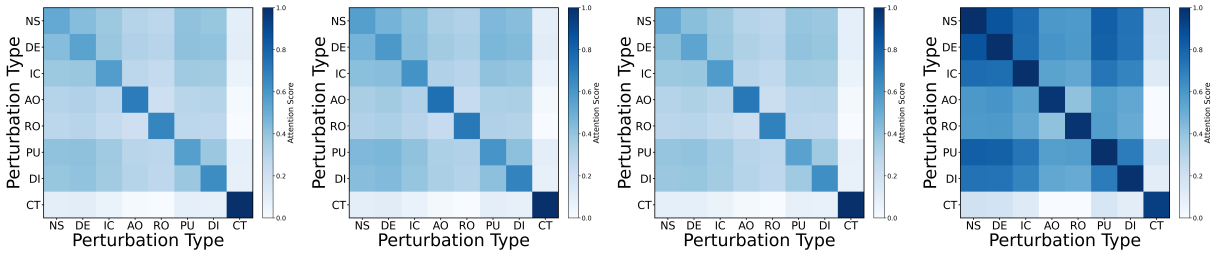
(a) layer 17 to layer 20



(b) layer 21 to layer 24

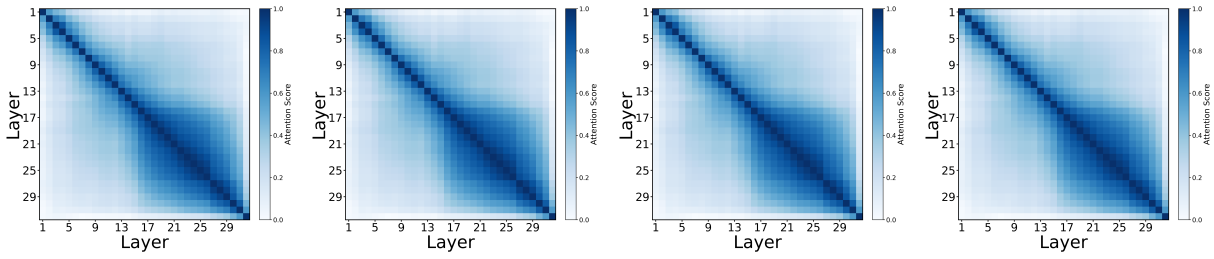


(c) layer 25 to layer 28



(d) layer 29 to layer 32

Figure 12: Cross-perturb activation heatmap for layer 17 to layer 32. We conduct experiments on the GSM-Plus benchmark with the Llama3.1-8B-Instruct model.



(a) Numerical Substitution (b) Digit Expansion (c) Inter-decimal-fraction Conversion (d) Adding Operation

(e) Reversing Operation (f) Problem Understanding (g) Distractor Insertion (h) Critical Thinking

Figure 13: Cross-layer activation heatmap for eight perturbations. We conduct experiments on the GSM-Plus benchmark with the Llama3.1-8B-Instruct model.