
Noise-Robustness Through Noise: A Framework combining Asymmetric LoRA with Poisoning MoE

Zhaokun Wang¹, Jinyu Guo^{1*}, Jingwen Pu¹, Lingfeng Chen¹,
Hongli Pu¹, Jie Ou¹, Libo Qin², Wenhong Tian^{1*}

¹School of Information and Software Engineering,
University of Electronic Science and Technology of China

²School of Computer Science and Engineering, Central South University
{guojinyu, tian_wenhong}@uestc.edu.cn

Abstract

Current parameter-efficient fine-tuning methods for adapting pre-trained language models to downstream tasks are susceptible to interference from noisy data. Conventional noise-handling approaches either rely on laborious data pre-processing or employ model architecture modifications prone to error accumulation. In contrast to existing noise-process paradigms, we propose a noise-robust adaptation method via asymmetric LoRA poisoning experts (LoPE), a novel framework that enhances model robustness to noise only with generated noisy data. Drawing inspiration from the mixture-of-experts architecture, LoPE strategically integrates a dedicated poisoning expert in an asymmetric LoRA configuration. Through a two-stage paradigm, LoPE performs noise injection on the poisoning expert during fine-tuning to enhance its noise discrimination and processing ability. During inference, we selectively mask the dedicated poisoning expert to leverage purified knowledge acquired by normal experts for noise-robust output. Extensive experiments demonstrate that LoPE achieves strong performance and robustness purely through the low-cost noise injection, which completely eliminates the requirement of data cleaning.

1 Introduction

Recently, pre-trained language models (PrLMs) have demonstrated remarkable success across various natural language processing tasks (Gao et al., 2025; Bianchi et al., 2024; Guo et al., 2025; Yang et al., 2025). To further enhance model performance on downstream tasks, researchers typically employ domain-specific corpora for targeted fine-tuning of pre-trained models (Li et al., 2024a). However, this mainstream solution critically depends on the quality of training data in both pre-training and fine-tuning stages, though with distinct emphases. Different from the pre-training stage, where models acquire general linguistic knowledge through exposure to large-scale corpora, the fine-tuning stage requires models to adapt to specific downstream tasks using comparatively smaller datasets, thereby imposing stricter requirements on data quality (Mai et al., 2024; Shi and Lipani, 2023).

In downstream NLP applications (Ou et al., 2025), noise poses multiple critical challenges: 1) labeling errors, syntactic irregularities, and extraneous content disrupt the model’s ability to capture and learn effective features, leading to unstable training and hindering the establishment of precise, robust knowledge systems (Mao et al., 2023; Xu et al., 2024); 2) noise weakens model generalization when encountering unseen data, impairing flexible knowledge transfer (Ahn et al., 2024); 3) noisy data may introduce biases, causing models to disproportionately emphasize or neglect specific categories,

*Corresponding authors.

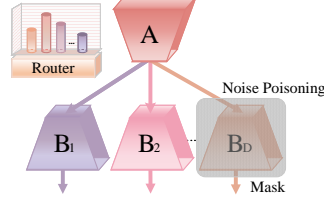


Figure 1: The unique configuration of the asymmetric LoRA architecture in our approach, where the grey area on the right represents the mask for the poisoning expert, thereby eliminating the knowledge affected by the noise learned by the poisoning expert.

compromising fairness and interpretability during fine-tuning stage (Zhao et al., 2024). These issues collectively manifest as drastic performance degradation in downstream tasks.

Recent research on noise can be broadly divided into two types of work. The first type focuses on reconstructing the data before training by cleaning, filtering, or relabeling to construct purified datasets (Ye et al., 2022; Yuan et al., 2024). The second type involves developing dedicated denoising architectures during training (Wu et al., 2023; Zhu et al., 2024). Through the analysis of existing denoising approaches, we identify two primary limitations in current research paradigms: (1) Numerous methods heavily rely on manual intervention or prior assumptions during data pre-processing, requiring noise detection and cleaning before model training; (2) Methods that focus on improving the model architecture during training avoid explicit data cleaning, but still cannot avoid discriminating noise information. In light of this, the effectiveness of these methods is often limited to specific noise types or data distributions. In addition, data processing pipelines not only incur additional computational and annotation expenses but also lead to error propagation.

Fortunately, compared to noise injection identification and processing, noise injection presents a cost-effective and easily automatable alternative that enhances data distribution authenticity through stochastic generation. However, noise addition to training data to improve model robustness seems like a pipe dream, as both clean and noisy samples influence all model parameters, preventing optimal utilization of noise patterns. Therefore, to exploit noise data for beneficial robust effects, a possible approach is to integrate noise injection into a dedicated noise-adaptive module. Subsequently, the module that has learned the noise patterns can be eliminated through complementary set operations, thereby effectively shielding the main network from noise perturbations. We name this strategy “Guided Poisoning”.

To achieve this goal, it is necessary to model and separate the noise with dedicated modules to split it. The recent emergence of Mixture-of-Experts (MoE) systems (Shazeer et al., 2017) provides an ideal implementation framework, as their architectural design naturally supports functional specialization through expert combinations. In this context, Low-Rank Adaptation (LoRA) (Hu et al., 2022) within asymmetric MoE architectures offers unique advantages. As shown in Figure 1, the modified MoE-LoRA architectures replace multiple A matrices with a shared A matrix, where each matrix B forms an independent expert. Prior studies (Tian et al., 2024) demonstrate that in such asymmetric LoRA architectures, the shared A matrix typically captures universal knowledge, while individual B matrices adapt to discrepancy knowledge.

Building on these insights, this paper proposes a noise-robust adaptation method via asymmetric **LoRA poisoning experts (LoPE)**. Specifically, we take one of the matrices B to serve as the poisoning expert within the MoE to route and process noise data as shown in B_D on the right side of Figure 1. During the inference stage, we achieve noise robustness by masking the pathway of the poisoning expert. The overall process consists of two stages: fine-tuning (I-I, I-II) and inference (II).

I-I) In the first stage of fine-tuning, we propose a hybrid noise injection (HyNoIse) approach that combines augmentation at both the input and embedding levels to enhance the original dataset. While freezing other parameters, we train the matrix A and poisoning expert B_D of LoRA with the augmented data. This strengthens the poisoning expert’s noise processing capability and leverages the properties matrix A to learn universal knowledge.

I-II) In the second stage of fine-tuning, we focus on the router’s ability to guide both clean and noisy data and the normal experts’ capacity for learning from clean data. Specifically, we fine-tune

the entire LoRA architecture using original data without noise injection, freezing the poisoning expert B_D 's parameters but allowing its participation in forward propagation. Moreover, to further strengthen the integrity of all expert outputs, we propose the Dynamically Compensated Expert Synergy (DyCompEnSate) mechanism to mitigate the impact of removing the poisoning expert during inference.

II) During inference, we only utilize the outputs from normal experts trained on the original data. By masking the poisoning expert, we complete the inference task with nearly noise-free knowledge.

Extensive experiments on benchmark datasets across mainstream NLP tasks demonstrate that our method achieves strong performance without relying on processed clean data, addressing limitations in existing denoising methods. A subsequent series of experiments further validates the fitness between our asymmetric LoRA architecture and noise-robust enhancement. Notably, since LoPE is based on a low-rank parameter-efficient fine-tuning framework, it maintains noise robustness while also achieving high efficiency. The main contributions of this paper are summarized as follows:

1. We propose LoPE, a novel noise-robust adaptation method that utilizes noise injection to handle noise. This approach integrates the noise handling module as an independent expert using the asymmetric LoRA architecture. Combined with multi-stage fine-tuning and masking operations, we enhance the model's robustness to noise only through the generated noisy data.
2. We design a flexible hybrid noise injection strategy, introducing discrete noise at the input level and continuous noise at the embedding level. We also devise a dynamically compensated expert synergy mechanism to compensate for the gap caused by directly masking the poisoning expert.
3. Extensive experiments on multiple mainstream benchmark datasets show that our method achieves competitive performance, which departs from conventional noise-handling paradigms and addresses critical limitations in existing denoising methods.

2 Related Work

2.1 Research on Data Noise

As data quality continues to play an increasingly crucial role in the performance of deep learning models, handling data noise has become a pressing and critical challenge across various domains. Recently, numerous studies have been conducted to address this issue, aiming to enhance the accuracy and reliability of deep learning models. The primary approach involves constructing denoising models for specific tasks or data structures to learn from the data. (Xu et al., 2024) designed MICL, a contrastive learning module to filter out irrelevant interactions in recommendation systems. (Biester et al., 2024) introduced LLMClean, leveraging generated contextual modeling and rule-based correction for tabular data denoising. (Mao et al., 2023) proposed LeCoRE, which mainly solves the problem of complex search intent understanding in conversational search. These existing methods usually have limitations when dealing with data from different domains, and the trained models are difficult to transfer to other domains. Another common approach involves cleaning, filtering, or relabeling raw data before training to eliminate noisy samples and construct a cleaner dataset. (Feng et al., 2024) introduced an intelligent receiver integrating a pre-denoising network and an LSTM module for denoising preprocessing. (Ji et al., 2024) developed a self-denoising method to refine LLM predictions on noisy inputs. Despite the advancement of existing denoising approaches, they have certain limitations, such as essential noise detection, limitation to data distributions, and error propagation cascades.

2.2 Development of LoRA

LLMs have attracted considerable attention due to their powerful natural language processing capabilities (Kenton and Toutanova, 2019). However, the massive parameters of these models make full-parameter tuning prohibitively expensive. Low-Rank Adaptation (LoRA) (Hu et al., 2022) decomposes weight updates into trainable low-rank matrices (A , B), maintaining performance while drastically reducing parameters and computation, establishing it as a key parameter-efficient fine-tuning (PEFT) (Li and Liang, 2021) technique. Recent innovations integrate LoRA with Mixture-of-Experts (MoE) architectures. MoE-LoRA (Dou et al., 2024) synthesizes complementary strengths: MoE's specialized expert networks with gating mechanisms synergize with LoRA's low-rank adap-

tations, enabling cross-domain adaptability. This hybrid approach mitigates multitask interference while preserving parameter efficiency through task-specific adaptor coordination. (Li et al., 2024b) integrated LoRA experts into the feedforward network (FFN) layers of a frozen pre-trained model, improving task adaptability while addressing imbalance issues in MoE models. To address the performance degradation of LoRA in complex corpora while maintaining parameter efficiency, (Tian et al., 2024) developed HydraLoRA, an asymmetric structure that partitions LoRA modules into specialized experts with MoE-based routing, improving efficiency and adaptability in heterogeneous corpora. The asymmetric LoRA structure’s unique design enables explicit noise modeling and separates noise-processing modules. This provides a foundation framework for achieving noise robustness through the generated noise.

3 Methodology

In this section, we present a comprehensive description of our proposed noise-robust adaptation method via asymmetric LoRA poisoning experts, which encompasses two stages: fine-tuning and inference, as illustrated in Figure 2. We begin by introducing the asymmetric LoRA architecture and the construction of poisoning experts, followed by a sequential explanation of our framework’s workflow. Details of each module are given respectively in the remainder of this section.

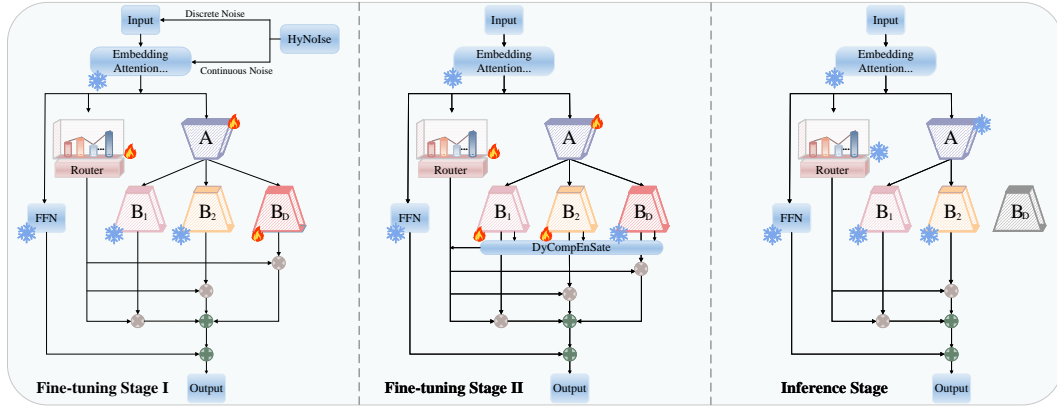


Figure 2: The LoPE pipeline consists of two stages: fine-tuning and inference stages. Fine-tuning Stage I: HyNoise is to enhance the poisoning expert’s noise understanding. Fine-tuning Stage II: Freezing the poisoning expert while fine-tuning the remaining experts and the shared matrix. Inference stage: Masking the poisoning expert with noise-affect knowledge, allowing the remaining experts to generate outputs that are relatively robust to noise.

3.1 Asymmetric LoRA architecture

Here we first introduce the asymmetric LoRA architecture (Tian et al., 2024) we employed. This architecture utilizes a centrally shared matrix A and multiple distinct matrices B_i , shown as follows: $W = W_0 + \sum_{i=1}^N \omega_i B_i A$. Compared to symmetric architectures, asymmetric configurations demonstrate superior advantages in mitigating parameter redundancy and enhancing computational efficiency. By sharing a central matrix A , the architecture significantly reduces learnable parameters while maintaining representational capacity, yielding a more lightweight architecture. The implementation of multiple matrices B_i enables flexible assignment of specialized adaptive modules for distinct objectives.

Because of these relatively independent experts, our approach, which involves isolating a dedicated module within the model to separate and process noise, becomes technically feasible. In LoPE, the matrix B is treated as the expert adaptor. Furthermore, a randomly selected matrix B is designated as a dedicated poisoning expert, denoted as B_D . This expert shares the same structure as the other experts but undergoes a targeted modification of its capabilities through the process described in the following sections, with the aim of identifying and processing noise. Its forward propagation is as follows: $W = W_0 + (\omega_D B_D + \sum_{i=1}^{N-1} \omega_i B_i) A$.

3.2 Fine-tuning Stage

3.2.1 Hybrid Noise Injection (HyNoIse)

The original dataset is defined as $S = (x_i, p_i)_{i=1}^M$, where x_i denotes model inputs and p_i their corresponding target labels. To increase the noise ratio in the data, we implement a hybrid data augmentation strategy that integrates both discrete and continuous noise perturbations, with multi-level noise coverage spanning character, token, label, and structural anomalies. To uniformly cover disturbances from all noise types, we adopt an equal-injection strategy. First, discrete noise injection is performed using a function $\text{NoiseFunction}(\cdot)$. Concretely, we adopt three representative and commonly referenced noise strategies that randomly introduce character-level noise (including word order shuffling, noise character insertion, and character deletion) to generate modified inputs, x'_i . These transformations enable to simulation of realistic disturbances such as character-level corruption and random truncations. The augmented dataset is then produced as $S' = (x'_i, p'_i)_{i=1}^M$.

Additionally, we inject continuous noise at the embedding level, influencing both the label and token levels. Inspired by (Jain et al., 2023), let $E \in \mathbb{R}^{b \times l \times d}$ represent the embedding for input sequences, where b , l , and d correspond to batch size, sequence length, and embedding dimension, respectively. We introduce an attention mask $M \in \mathbb{R}^{b \times l \times 1}$ indicating valid token positions and continuous noise $N \sim \mathcal{U}(-1, 1)^{b \times l \times d}$ from a uniform distribution. The noise-injected embeddings are computed as $E' = E + \alpha N \odot M$, where \odot denotes element-wise multiplication, and α stands for noise ratio. Multiplying the noise with the attention mask element-wise ensures that the noise is applied only to the valid positions of the sequence. The proposed HyNoIse method combines these complementary noise injection strategies to expose the model to diverse noise patterns during fine-tuning.

3.2.2 Fine-tuning Stage I: Specialized Poisoning Expert

In Stage I, we exclusively fine-tune the poisoning expert B_D and shared matrix A using the HyNoIse method from Section 3.1 while keeping the other expert adapters B_i frozen. This stage approach enables B_D focused learning of noise-handling without cross-expert interference. The forward computation during this stage is formalized as:

$$y = W_0 x + (\omega_D B_D + \sum_{i=1}^{N-1} \omega_i f(B_i)) A x, \quad (1)$$

where ω_D denotes the adaptive weighting coefficient for the poisoning expert, which is part of ω_i , and $f(\cdot)$ stands for freezing parameter operation.

3.2.3 Fine-tuning Stage II: Dynamically Compensated Expert Synergy

In Stage II, we achieve collaboration and complementarity among all expert adapters. To simultaneously enhance the routing mechanism’s discriminative capacity for noise patterns and optimize normal experts for clean data processing, we conduct joint fine-tuning of all adapter components while preserving the poisoning expert’s specialization. During this phase, we maintain parameter freezing of the B_D from Stage I to keep its capabilities, while updating the remaining expert matrices B_i and shared matrix A . The forward computation evolves to:

$$y = W_0 x + (\omega_D f(B_D) + \sum_{i=1}^{N-1} \omega_i B_i) A x. \quad (2)$$

We introduce a router network (gating function) to dynamically allocate expert adapters’ contributions. It processes the token representation x through a fully connected layer and softmax to compute expert weights $\omega_i = \text{softmax}(W_{\text{gate}}^T x)$, where $W_{\text{gate}} \in \mathbb{R}^{r \times N}$ denotes the router’s parameters. To address the interdependency disruption caused by expert masking in the inference stage, we propose the Dynamically Compensated Expert Synergy (DyCompEnSate) mechanism. Our analysis reveals that joint expert training establishes learned inter-expert correlations, where the direct mask of B_D during inference has the potential to disrupt these dependencies. DyCompEnSate dynamically preserves these relationships through dependency-aware weight compensation. Formally, we construct an expert dependency matrix $\theta \in \mathbb{R}^{N \times N}$, where the element θ_{ij} quantifies the similarity of output between experts i and j . The poisoning expert dependencies θ_{iD} are computed via output similarity

analysis: $\theta_{iD} = \text{sim}(y_i, y_D)$, $i = 1, 2, \dots, N, i \neq D$, where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity metric. These dependencies compensate expert contributions through dynamic weight amplification:

$$y = W_0x + \beta(\omega_D f(B_D) + \sum_{i=1}^K (1 + \theta_{iD}) \omega_i B_i) Ax, \quad (3)$$

where K indicates the number of active experts participating in the computation. To maintain output stability, we introduce β to adjust the weight parameter normalization.

3.3 Inference Stage

During the inference stage, we exclusively employ experts that have acquired purified knowledge during the fine-tuning stage, which naturally leads to masking the poisoning expert while preserving the normal experts. Since the matrices B_i play the role of linear transformations, in the initial stage, we calculate the weighted average of the remaining experts to aggregate their specialized knowledge. Based on this, we perform the parameter-efficient fine-tuning transformation on the input data using the aggregated expert knowledge. In the inference stage, all parameters are frozen, and the contribution of expert adapters is dynamically adjusted based on the input data, thereby achieving the merging of adapters. Specifically, the poisoning expert B_D is masked, such that the system exclusively outputs inference results from the remaining experts trained on clean data to achieve knowledge purification. The result obtained after inference:

$$y = W_0x + \beta \sum_{i=1}^K (1 + \theta_{iD}) \omega_i f(B_i) Ax. \quad (4)$$

4 Experiments

4.1 Datasets, Evaluation Metrics and Compared Baseline Models

Datasets and Evaluation Metrics. The dataset used in our experiment is from Alpaca-52K (Taori et al., 2023), which consists of 52K instruction-following data. We used 10% of the dataset for fine-tuning. The dataset spans a wide range of task types, including calculations, question answering, and creation, among others. For evaluation, we first utilized the Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020) benchmark, and additionally selected three domains in MMLU encompassing a total of 30 distinct tasks, for in-depth analysis: Natural Sciences and Engineering Technology (NSET), Social Behavior and Humanities (SBH), and History. We also use the GSM8K (Cobbe et al., 2021) math problem set, Question Answering datasets (Physical Interaction QA (Bisk et al., 2020), Social Interaction QA (Sap et al., 2019), and ARC-easy (Clark et al., 2018)). We used different datasets for fine-tuning and evaluating, which effectively mitigates the issue of domain shift. Accuracy (%) is used as the evaluation metric across all datasets. This setup systematically evaluates the model’s noise-robust capabilities across general scenarios and specialized domains. A detailed description of these datasets can be found in Appendix A.

Baselines. LoPE is a general fine-tuning framework that works across different noise environments and data types, and does not conflict with existing denoising methods. Therefore, we mainly compare it with similar fine-tuning methods. Specifically, we compared LoPE with five baseline PEFT methods on the same datasets: (1) P-Tuning (Liu et al., 2024), (2) Prefix Tuning (Li and Liang, 2021), (3) AdaLoRA (Zhang et al.), (4) LoRA (Hu et al., 2022), and (5) HydraLoRA (Tian et al., 2024). Detailed descriptions of these methods are provided in Appendix B.1.

4.2 Implementation Details

We define two modes for the fine-tuning datasets: *Orig*, corresponding to the publicly available version discussed in Section 4.1, and *Nois*, the version with discrete noise injected. To simulate the real-world noise distributions and create a challenging and impartial noise learning environment across all baseline models, we employed a uniform discrete noise injection strategy same as Section 3.2.1 to generate the *Nois* dataset, introducing a controlled proportion of noise into the original data. Empirical studies (Song et al., 2022) suggest real-world datasets contain 8%-38.5% noise. Given that

Table 1: The comparative performance (%) of different PEFT methods across various datasets. The methods marked with a † are fine-tuned with the *Nois* datasets, while all subsequent methods are fine-tuned based on *Orig* datasets. r denotes the rank, #A represents the number of shared matrices, #B represents the number of *B* matrices involved in the inference stage.

Methods	MMLU	PIQA	SIQA	GSM8K	ARC-e	NSET	SBH	History	%Param	#A	#B
HydraLoRA(r=4)	46.10	76.28	52.92	16.15	62.61	35.99	54.78	55.95	0.062	1	3
LoPE(r=4)	46.86	76.99	53.58	17.89	64.20	36.82	54.46	56.66	0.062	1	3
P-Tuning†	37.23	71.65	39.97	8.87	45.21	26.04	37.14	45.09	0.193	-	-
Prefix Tuning†	37.91	71.79	40.42	9.25	43.48	27.13	38.77	44.21	0.077	-	-
AdaLoRA(r=2)†	39.11	72.29	41.07	10.38	47.84	29.16	39.83	49.69	0.023	1	1
LoRA(r=2)†	38.22	69.47	40.94	9.13	46.24	26.17	38.61	46.83	0.015	1	1
LoRA(r=4)†	40.45	71.45	43.17	11.02	48.39	27.96	40.88	49.03	0.031	1	1
HydraLoRA(r=2)†	42.47	74.65	46.38	10.74	56.08	31.26	42.37	52.65	0.031	1	3
HydraLoRA(r=4)†	43.08	74.92	47.29	11.83	56.26	34.74	52.35	55.80	0.062	1	3
HydraLoRA(r=8)†	43.98	75.30	48.12	12.44	57.47	35.23	53.07	56.35	0.124	1	3
LoPE(r=2)†	43.05±0.28	75.03±0.30	46.76±0.17	11.23±0.37	58.93±0.51	33.36±0.47	48.65±0.32	54.72±0.29	0.031	1	3
LoPE(r=4)†	43.76±0.20	75.49±0.41	48.33±0.30	12.72±0.33	58.66±0.46	34.06±0.11	48.98±0.37	55.46±0.18	0.047	1	2
LoPE(r=4)†	44.42±0.18	76.28±0.38	49.03±0.41	13.72±0.34	60.49±0.27	35.45±0.33	52.88±0.20	56.84±0.46	0.062	1	3
LoPE(r=8)†	44.82±0.24	76.83±0.35	49.90±0.27	14.31±0.38	62.36±0.47	35.79±0.13	53.71±0.32	58.02±0.34	0.124	1	3

the original dataset already contains some inherent noise, we used a conservative injection rate of 5% to ensure consistency across models and maintain typical noise levels. The HyNoIse ratio α used in our main experiment is also set to 5%, and the number of poisoning experts is set to 1. Detailed descriptions of experiment configurations are provided in Appendix B.2.

4.3 Main Results

In this section, we evaluate LoPE’s effectiveness through extensive experiments and compare it with several mainstream PEFT methods on LLaMA2-7b (Touvron et al., 2023). Table 1 presents the performance of LoPE alongside the baseline methods. MMLU represents the average accuracy across all tasks. NSET, SBH, and History represent the average accuracy of tasks within each respective set. The specific accuracy for each sub-task can be found in Appendix D.2. We first compared LoPE with HydraLoRA on the *Orig* datasets mentioned in Section 4.2. In the original environment, LoPE exhibited only a limited improvement across these datasets. Given that the *Orig* datasets are nearly noise-free with minimal inherent noise, LoPE’s noise adaptation capability was not significantly demonstrated. Subsequently, we conducted experiments on the *Nois* datasets. The results indicate that LoPE exhibits robustness and adaptability across most tasks. Notably, LoPE excelled across three QA tasks, achieving a 4.89% improvement on the ARC-e with a rank of 8. Consistent with our theoretical analysis, LoPE also showed an improvement on the GSM8K task. Moreover, LoPE outperformed baseline methods in the MMLU benchmark (average accuracy of 57 sub-tasks), achieving a 1.34% average improvement over HydraLoRA with a rank of 4. Specifically, as the sub-task of MMLU, NSET, similar to GSM8K, often involves precise computations, LoPE maintained coherent reasoning under noise, whereas traditional methods struggled. Furthermore, LoPE also outperforms other methods in complex tasks such as SBH and History, which involve long texts and logical reasoning. In conclusion, LoPE overcomes the limitations of conventional denoising methods, achieving enhanced robustness through noise injection. Notably, our method maintains the same time complexity of $O(n^2)$ as other fine-tuning methods, ensuring high computational efficiency. In Table 1, we also explore LoPE’s sensitivity to parameter configurations by examining the impact of varying the rank and the number of experts, as well as comparing its parameter scale with other methods. Higher ranks generally lead to better performance; However, increasing the rank also incurs higher parameter and computational costs, thus requiring a balance between scalability and effectiveness. In the following experiments, the rank is set to 4 by default, with three normal experts and one poisoning expert.

4.4 Ablation Study

4.4.1 Ablations on HyNoIse

In this section, we explore the effectiveness of the proposed HyNoIse method. We fine-tune the model on *Nois* data and set different basic noise ratios (3.5%, 5%, 8%). At the same time, the

ratio of our HyNoIse method is 5%, and the results are shown in Table 2. The results indicate that both discrete noise and continuous noise can improve the method. Our method maintains relatively stable performance when handling different levels of noise. To further verify the effectiveness of our proposed HyNoise method, we conduct ablation studies by selectively removing each type of noise (discrete noise or continuous noise) during injection. As evidenced in Table 2, the robustness of the model is somewhat reduced under single-noise conditions. However, performance with either discrete or continuous noise still surpasses the None type noise injection, thereby highlighting the effectiveness of both noise types. Notably, fine-tuning with discrete-only noise outperforms fine-tuning with continuous-only noise. This can be attributed to the fact that discrete noise directly manipulates the original natural language text, facilitating more effective alignment in the semantic space during the fine-tuning stage II.

Table 2: Average accuracy (%) of LoPE on PIQA and SIQA datasets under different noise conditions. None indicates that no noise is injected during the fine-tuning stage I.

Noise Type	None	Continuous	Discrete	Hybrid
3.5% Level	60.90	61.23	62.07	63.31
5% Level	59.89	60.71	61.95	62.66
8% Level	57.48	58.68	60.14	61.86

To further investigate whether LoPE’s strong performance depends on the consistency between the noise types in the *Nois* dataset and the discrete noise in HyNoIse, we conducted additional experiments introducing inconsistent noise injection between the dataset construction and the discrete noise in HyNoIse. The results in Table 3 show that when different operations are applied for LoPE training and dataset construction, all LoPE variants achieve considerable improvements over the baseline HydraLoRA. This demonstrates that LoPE’s effectiveness generalizes across different noise categories and does not strongly depend on noise type consistency.

Moreover, the observed performance gains on the *Orig* dataset (Table 1) further indicate that LoPE is effective and generalizable to naturally occurring noisy data. It is worth noting that HyNoIse also injects continuous noise, which contributes to enhancing the model’s robustness under multiple types of noise.

Table 3: Performance comparison of LoPE under **inconsistent noise injection** and HydraLoRA. Each row represents a distinct noise operation. Abbreviations used: NCI (Noise Character Insertion), WOS (Word Order Shuffling), CD (Character Deletion), and WR (Word Replacement), representing four types of noise operations.

Model	Dataset	HyNoIse	MMLU	PIQA	SIQA	GSM8K
LoPE	Orig + NCI	Continuous + WOS	44.58	75.15	50.11	14.92
HydraLoRA	Orig + NCI	None	44.24	75.03	49.75	12.78
LoPE	Orig + WOS	Continuous + CD	45.09	75.84	50.55	14.64
HydraLoRA	Orig + WOS	None	44.37	75.21	49.65	12.34
LoPE	Orig + WR	Continuous + NCI	45.13	75.35	50.17	14.73
HydraLoRA	Orig + WR	None	44.31	75.07	49.73	12.39

We further applied HyNoIse directly to the original HydraLoRA framework (Table 4) and observed a notable performance drop. This is mainly due to the absence of the division of labor between normal experts and poisoning experts, and the multi-stage differentiated training, causing noisy and clean data to equally affect the model from the start. Therefore, LoPE’s key contribution is its architecture and multi-stage process, enabling explicit noise filtering and robustness.

4.4.2 Impact of Backbone LLMs

To evaluate the impact of backbone architectures on LoPE’s performance, we conducted model substitution experiments, with results summarized in Table 5. LoPE consistently demonstrates stable performance improvements across LLaMA2-7b (a decoder-only model), T5-large (a classic encoder-decoder model), as well as Qwen2-7b and Qwen1.5-14b. This result indicates that LoPE’s

Table 4: Performance comparison of LoPE and HydraLoRA (with/without HyNoIse) across benchmarks. The methods marked with a \dagger are fine-tuned with the *Nois* datasets.

Methods	MMLU	PIQA	SIQA	GSM8K
HydraLoRA \dagger (with HyNoIse)	40.57	73.40	45.03	8.47
HydraLoRA \dagger	43.08	74.92	47.29	11.83
LoPE \dagger	44.42	76.28	49.03	13.72

performance gains primarily stem from its asymmetric LoRA architecture and two-stage training strategy, which strengthen its generalization across diverse tasks and application scenarios.

Table 5: Performance (%) of different backbone LLMs of our method on SIQA dataset after fine-tuning of *Nois* datasets (5%).

Approaches	T5-large	LLaMA2-7b	Qwen2-7b	Qwen1.5-14b
HydraLoRA	33.64	47.29	66.49	78.23
LoPE	36.37	49.03	68.99	80.02

4.4.3 Ablation Study on DyCompEnSate

We verified the effectiveness of the proposed DyCompEnSate method. First, we directly masked the output of B_D without introducing the DyCompEnSate mechanism, as shown in Figure 3, when the column number = 0, which indicates that no experts are involved in the DyCompEnSate method, accuracy dropped at the lowest value.

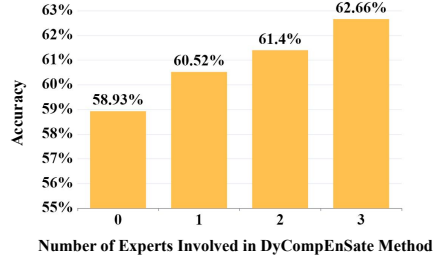


Figure 3: The relationship between the number of experts involved in the DyCompEnSate method and the average accuracy (%) of datasets PIQA and SIQA.

This phenomenon validates our hypothesis: when all experts are jointly trained, there exist significant inter-dependencies among them, and directly eliminating the output of B_D indirectly affects other experts' outputs. With the introduction of DyCompEnSate and the increase in the number of normal experts involved, it dynamically weights the amplification of the contribution of other experts to mitigate the information loss caused by removing B_D in a smooth and adaptive manner.

These results demonstrate the existence of inter-expert dependencies and the effectiveness of our compensation strategy in preserving performance when masking a compromised expert.

5 Analysis

5.1 Does Higher HyNoise Ratio Enhance Performance?

In Section 4.2, while our baseline experiments employed a 5% HyNoIse ratio, we investigated whether an increased noise ratio during Stage I enhances noise robustness. We evaluated four HyNoIse ratios (3.5%, 5%, 10%, 15%), with comparative results visualized in Figure 4. Contrary to potential expectations, our analysis reveals a non-monotonic relationship between HyNoIse intensity and model robustness. Excessive noise injection (i.e. $\geq 10\%$) degrades the router's discriminative capability for noise patterns, as evidenced by reduced expert specialization. We consider that this was due to overwhelming data distortion, which prevents effective differentiation between clean and

noisy inputs, ultimately compromising expert weight allocation. We also evaluated the performance under different basic noise ratios; additional experimental results are provided in Appendix D.1.

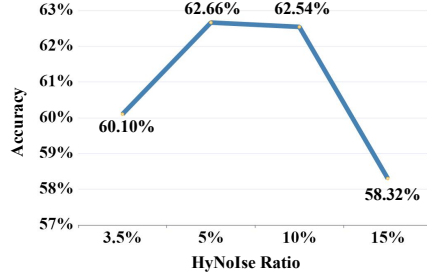


Figure 4: Average accuracy to different HyNoise ratios in PIQA and SIQA datasets.

5.2 Can the Poisoning Expert Truly Accomplish Its Task?

The core of our proposed method is the pioneering introduction of the poisoning expert. In Section 4.3, we investigated the impact of the number of normal experts in LoRA on the experimental results. Here, we discuss the influence of the number of poisoning experts on the results. According to the design, poisoning experts play an auxiliary role. Therefore, their number should be lower than normal experts. We conducted experiments with three different configurations of the number of experts, and the parameter freezing settings remain consistent with Section 3. The experimental results in Table 6 demonstrate that increasing the number of poisoning experts has little impact, as it does not affect the objective of stage I of fine-tuning. These poisoning experts share the functionality previously assigned to a single poisoning expert. During stage II of fine-tuning, after freezing their parameters, the normal experts can learn clean knowledge as usual. When the total number of experts remains constant, changing the number of poisoned experts will have a significant impact on the results, as the normal experts responsible for inference are replaced.

Table 6: Average accuracy of LoPE in PIQA and SIQA datasets with different experts, PE is the number of poisoning experts, NE is the number of normal experts.

Method	NE=3,PE=1	NE=3,PE=2	NE=2,PE=2	Mask(NE=3,PE=1)	Not Mask(NE=3,PE=1)
LoPE	62.66	62.31	60.31	62.66	60.75

Also, we propose that the involvement of the poisoning expert in the inference process may result in the contamination of knowledge to a catastrophic degree. In light of this, we set up a comparative experiment: Masking vs. Not Masking. The experimental results show that after the first-stage fine-tuning, the poisoning expert intends to specifically adapt and handle the noisy data. If the poisoning expert’s output is not masked during inference, allowing it to participate, the model may be affected by the misinformation it learns from the noise, which may affect the reliability of the inference results. This aligns with our original intention for designing the poisoning expert. By masking the poisoning expert that learns from noise-affected data during inference, this approach allows other experts to perform inference using relatively pure knowledge.

6 Conclusion

In this paper, we introduce LoPE, a noise-robust adaptation method via asymmetric LoRA poisoning experts, which is specifically designed to address performance degradation caused by noisy datasets. The two-stage paradigm and selective masking mechanism enable the model to enhance its robustness only through generated noise perturbations, as evidenced by our experimental results. This approach effectively overcomes the inherent limitations of conventional noise-handling methods. In future research, we believe the combination of the two-stage fine-tuning paradigm and selective masking mechanism opens a door to a promising area of intractable problem resolution through complement set approaches, and we will explore it for more than noise processing in the future.

Acknowledgments and Disclosure of Funding

This research is supported by the Sichuan International Science and Technology Innovation Cooperation Project (also known as the Hong Kong, Macao, and Taiwan Science and Technology Innovation Cooperation Project, ID: 2024YFHZ0317), the Chengdu Science and Technology Bureau Project (ID: 2024-YF09-00041-SN), the National Natural Science Foundation of China (Project ID: W2433163), Huawei Funding (Project ID: H04W241592), and the Postdoctoral Fellowship Program (Grade C) of the China Postdoctoral Science Foundation (Grant No. GZC20251053).

References

Qwen2 technical report. 2024.

Sumyeong Ahn, Sihyeon Kim, Jongwoo Ko, and Se-Young Yun. Fine-tuning pre-trained models for robustness under noisy labels. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 3643–3651. International Joint Conferences on Artificial Intelligence Organization, 2024. Main Track.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

F Bianchi, P Chia, M Yuksekgonul, J Tagliabue, D Jurafsky, J Zou, et al. How well can llms negotiate? negotiationarena platform and analysis. *PROCEEDINGS OF MACHINE LEARNING RESEARCH*, 235:3935–3951, 2024.

Fabian Biester, Mohamed Abdelaal, and Daniel Del Gaudio. Llmclean: Context-aware tabular data cleaning via llm-generated ofds. In *European Conference on Advances in Databases and Information Systems*, pages 68–78. Springer, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7432–7439, 2020.

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

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

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, 2024.

Yunqi Feng, Yunxian Chen, Huimei Han, Weidang Lu, Nan Zhao, Arumugam Nallanathan, and Xiaoni Yang. Deep learning based physical-layer receiver with pre-denoise. *IEEE Transactions on Vehicular Technology*, 2024.

Jianqi Gao, Jian Cao, Ranran Bu, Nengjun Zhu, Wei Guan, and Hang Yu. Promoting knowledge base question answering by directing llms to generate task-relevant logical forms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23914–23922, 2025.

- Jinyu Guo, Xunlei Chen, Qiyang Xia, Zhaokun Wang, Jie Ou, Libo Qin, Shunyu Yao, and Wenhong Tian. HASH-RAG: Bridging deep hashing with retriever for efficient, fine retrieval and augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26847–26858, 2025.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuezhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, et al. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*, 2023.
- Jiabao Ji, Bairu Hou, Zhen Zhang, Guanhua Zhang, Wenqi Fan, Qing Li, Yang Zhang, Gaowen Liu, Sijia Liu, and Shiyu Chang. Advancing the robustness of large language models through self-denoised smoothing. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 246–257, 2024.
- Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*. Minneapolis, Minnesota, 2019.
- Chengzhengxu Li, Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Chen Liu, Yu Lan, and Chao Shen. Concentrate attention: Towards domain-generalizable prompt optimization for language models. *Advances in Neural Information Processing Systems*, 37:3391–3420, 2024a.
- Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024b.
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.
- Zheda Mai, Arpita Chowdhury, Ping Zhang, Cheng-Hao Tu, Hong-You Chen, Vardaan Pahuja, Tanya Berger-Wolf, Song Gao, Charles Stewart, Yu Su, et al. Fine-tuning is fine, if calibrated. *Advances in Neural Information Processing Systems*, 37:136084–136119, 2024.
- Kelong Mao, Hongjin Qian, Fengran Mo, Zhicheng Dou, Bang Liu, Xiaohua Cheng, and Zhao Cao. Learning denoised and interpretable session representation for conversational search. In *Proceedings of the ACM Web Conference 2023*, pages 3193–3202, 2023.
- Jie Ou, Jinyu Guo, Shuaihong Jiang, Zhaokun Wang, Libo Qin, Shunyu Yao, and Wenhong Tian. Accelerating adaptive retrieval augmented generation via instruction-driven representation reduction of retrieval overlaps. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26983–27000, Vienna, Austria, 2025. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, 2019.

- Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.
- Zhengxiang Shi and Aldo Lipani. Don't stop pretraining? make prompt-based fine-tuning powerful learner. *Advances in Neural Information Processing Systems*, 36:5827–5849, 2023.
- Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153, 2022.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.
- Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. In *Advances in Neural Information Processing Systems*, pages 9565–9584. Curran Associates, Inc., 2024.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Shuhui Wu, Yongliang Shen, Zeqi Tan, Wenqi Ren, Jietian Guo, Shiliang Pu, and Weiming Lu. Mproto: Multi-prototype network with denoised optimal transport for distantly supervised named entity recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2361–2374, 2023.
- Fuyong Xu, Zhenfang Zhu, Yixin Fu, Ru Wang, and Peiyu Liu. Collaborative denoised graph contrastive learning for multi-modal recommendation. *Information Sciences*, 679:121017, 2024.
- Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. Moose-chem: Large language models for rediscovering unseen chemistry scientific hypotheses. In *International Conference on Representation Learning*, pages 33251–33277, 2025.
- Fanghua Ye, Yue Feng, and Emine Yilmaz. Assist: Towards label noise-robust dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2719–2731, 2022.
- Bo Yuan, Yulin Chen, Yin Zhang, and Wei Jiang. Hide and seek in noise labels: Noise-robust collaborative active learning with llms-powered assistance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10977–11011, 2024.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.
- Jinxu Zhao, Guanting Dong, Yueyan Qiu, Tingfeng Hui, Xiaoshuai Song, Daichi Guo, and Weiran Xu. Noise-bert: A unified perturbation-robust framework with noise alignment pre-training for noisy slot filling task. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6150–6154. IEEE, 2024.
- Kun Zhu, Xiaocheng Feng, Xiyuan Du, Yuxuan Gu, Weijiang Yu, Haotian Wang, Qianglong Chen, Zheng Chu, Jingchang Chen, and Bing Qin. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1069, 2024.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: The Abstract Section and Section 1 of this article accurately reflect the contribution and scope of the paper.

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 discussed the limitations of this work in Appendix C.

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: [\[Yes\]](#)

Justification: The paper provides the full set of assumptions and a complete (and correct) proof.

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: The paper fully discloses 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.

Guidelines:

- The answer NA means that the paper does not include experiments.
- 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: The paper provides open access to the data and code.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We have provided relevant details in Section 4 and Section B.1.

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: [\[Yes\]](#)

Justification: We have provided relevant details in Section 4.

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 have provided relevant details in Section B.1

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.
- 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 research conducted in the paper conforms, in every 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: [Yes]

Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed in Section 6 and Section C

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: This paper does not address this issue.

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

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: [\[Yes\]](#)

Justification: This paper does not address this issue.

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: [\[Yes\]](#)

Justification: This paper does not address this issue.

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: [\[Yes\]](#)

Justification: This paper does not address this issue.

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: This paper does not address this issue.

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.

Appendix

A Dataset

A.1 Fine-tuning Dataset

Alpaca 52k is a dataset consisting of 52,000 instruction-following data, generated by Stanford University using the self-instruct method by the text-davinci-003. This dataset aims to enhance the model’s ability to understand and follow instructions. The dataset covers a wide range of domains and task types, including text generation, reasoning and explanation, transformation and calculation, classification and induction, question answering and suggestions, description and explanation, functionality and logic, as well as text processing and editing. By fine-tuning on these tasks, the model is better equipped to understand and execute various instructions, thereby improving its effectiveness and reliability in real-world applications.

A.2 Evaluation Dataset

Additionally, to comprehensively evaluate the model’s performance across different domains, we employed multiple evaluation benchmarks. To assess general-domain capabilities, we utilized the Massive Multitask Language Understanding (MMLU) benchmark. MMLU spans tasks across multiple fields, enabling a thorough evaluation of the model’s language understanding abilities. For our evaluation, we focused on three major categories: Basic Natural Sciences and Engineering Technology (NSET), Social Behavior and Humanities (SBH), and History. The specific tasks are listed in the table 7 below.

Table 7: Task categories and corresponding tasks

Task Category	Tasks
NSET	College Biology, College Chemistry, College Computer Science, College Mathematics, College Physics, Electrical Engineering, Abstract Algebra, Astronomy, Machine Learning, Computer Security, High School Chemistry, Elementary Mathematics, High School Physics, High School Computer Science, High School Statistics, High School Biology, High School Mathematics, Conceptual Physics.
SBH	Moral Disputes, Moral Scenarios, Professional Psychology, World Religions, High School Psychology, Philosophy, High School Government and Politics, Sociology.
History	Prehistory, High School US History, High School World History, High School European History.

NSET consists of 18 tasks related to science, engineering, technology, and mathematics. This category encompasses foundational disciplines within the natural sciences and engineering fields, including biology, chemistry, computer science, mathematics, physics, electrical engineering, astronomy, and more. These tasks cover biological knowledge, ranging from cell structure and molecular biology to ecology. In chemistry, tasks span from analytical chemistry to organic chemistry, covering fundamental laws (e.g., Newtonian mechanics and principles of chemical reactions). In computer science, research focuses on algorithms, graph theory, recursive methods, and the logic of technical implementation (e.g., algorithm design and circuit systems). Mathematical tasks encompass various branches, such as calculus, combinatorics, and ordinary differential equations. These tasks are highly sensitive to data and require significant specialization and logical reasoning, making them ideal for evaluating the model’s precision in computation and reasoning.

SBH includes 8 tasks from fields such as psychology, philosophy, sociology, and political science, focusing on human behavior patterns, moral decision-making, and ethical dilemmas within a societal context. The tasks involve analyzing moral controversies (e.g., freedom of speech, the death penalty, addiction), examining moral situations (e.g., violence, theft), exploring psychological principles behind individual behavior (e.g., personality development, emotional changes), and discussing ethical

dilemmas from a philosophical perspective (e.g., skepticism, utilitarianism). Additionally, the tasks address the impact of social structures on individual behavior, including socialization and inequality in sociology, and government functions and socio-political systems in political science.

The History category consists of 4 tasks related to the development of human history, including Prehistory, American history, world history, and European history. The content spans topics from the early evolution of humans and the origins of civilization to major historical events and transformations in modern and contemporary society. These tasks involve complex logical structures, reflecting the model’s depth of reasoning and its ability to transfer knowledge within the social sciences domain.

For evaluating mathematical reasoning ability, we adopted the GSM8K dataset, which includes complex mathematical problems and allows for a quantitative assessment of the model’s accuracy and efficiency in solving these problems.

The AI2’s Reasoning Challenge (ARC) dataset is a multiple-choice response dataset containing science exam questions for grades three through nine. The dataset is divided into two sections: an easy section (Arc-E) and a hard section (Arc-C) for questions that test the model’s ability in scientific reasoning. We used the Arc-E section.

Physical Interaction QA (PIQA) is a dataset designed to test a model’s ability to predict how objects interact in the physical world. The questions are typically based on everyday physical scenarios.

Social Interaction QA (SIQA) is a question-answering benchmark designed to assess a model’s ability to reason about social commonsense intelligence. SIQA focuses on inferring human actions and their social implications. The behaviors in SIQA span a wide range of social contexts, with answer choices including both human-crafted answers and adversarially filtered machine-generated options. Social IQa contains over 37,000 QA pairs, used to evaluate a model’s ability to reason about the social implications of everyday events and situations.

Through this multidimensional evaluation, we assessed the proposed method’s adaptability to both general and domain-specific tasks comprehensively, offering a thorough examination of the model’s capabilities across diverse multi-task settings and providing solid empirical support for future research.

B Baseline, Configurations and Backbone LLMs

B.1 Baseline

1. P-Tuning (Liu et al., 2024): P-Tuning employs a prompt encoder to automatically optimize continuous prompt embeddings, avoiding the limitations of manually designing discrete prompts. By flexibly inserting anchor tokens into the input sequence, P-tuning can effectively improve the alignment accuracy of the semantic space, achieving better task adaptation performance.
2. Prefix Tuning (Li and Liang, 2021): This method injects trainable prefix parameter matrices at the input of each layer of the Transformer. Through a hierarchical representation rectification mechanism, the prefix parameter matrices can guide the model output while keeping the backbone network parameters frozen. This approach allows for efficient adaptation to new tasks while retaining the pre-trained knowledge.
3. AdLoRA (Zhang et al.): Adaptive LoRA is based on a dynamic allocation algorithm that assesses parameter importance. It applies high-rank decomposition configurations to critical weight matrices and employs low-resource allocation strategies for non-core modules. This breaks the traditional paradigm of uniform parameter allocation, adaptively adjusting parameters for different parts to reduce computational and storage overhead while maintaining performance.
4. LoRA (Hu et al., 2022): LoRA (Low-Rank Adaptation) is an efficient fine-tuning technique for adapting pre-trained language models to downstream tasks. It aims to reduce the computational and storage overhead while maintaining the performance of the adapted model. The core idea of LoRA is to introduce low-rank update matrices to the pre-trained model. Instead of fine-tuning all the parameters of the model, LoRA keeps the original pre-trained weights frozen and injects trainable low-rank matrices into each layer of the model.
5. HydraLoRA (Tian et al., 2024): HydraLoRA is a parameter-efficient fine-tuning method based on the LoRA mechanism, which aims to improve the performance of large-scale language models (LLMs) in multi-task learning and cross-domain adaptation. This method involves the decomposition

of the LoRA module into multiple independent expert modules, thereby enhancing the original symmetrical LoRA structure into a single A and multi-B LoRA structure. This modification enables the model to respond to varying task requirements with greater flexibility.

B.2 Experiment Configurations

The experiment configurations are as follows: learning rate is 0.0002, random seed is 614, and epoch is 1. The hardware and software configurations used in our experiments are as follows.

CPU: Intel(R) Xeon(R) Platinum 8468V, 800MHZ, 48cores; GPU: NVIDIA TESLA H800 80 GB; Operating system: Ubuntu 20.04; Deep learning framework: Pytorch 1.13.1.

B.3 Backbone LLMs

1. T5-large (Raffel et al., 2020): T5-large is a pre trained language model based on the Transformer architecture, which has wide applications in the field of natural language processing. T5 stands for "Text-to-Text Transfer Transformer", which converts all NLP tasks into text-to-text format for processing. T5-large has over 7.7 million parameters, and compared to smaller T5 model variants, it is able to capture more complex patterns and semantic relationships in text. It performs well in tasks such as text generation, text classification, question answering systems, machine translation, etc. It can generate smooth and natural text, accurately understand context, and demonstrate good adaptability and universality in multiple language environments.

2. LLaMA2-7b: LLaMA2-7b is a large-scale language model launched by Meta, with approximately 7 billion parameters. It has excellent language comprehension and generation capabilities, and can handle various natural language tasks such as text generation, question answering, translation, summarization, etc. It performs well in multiple languages and can be widely applied in various scenarios, providing users with convenient text processing services. Its open-source and free commercial license characteristics also promote its application and research in many fields.

3. Qwen2-7b (qwe, 2024): Qwen2-7b is an open-source model launched by the Alibaba Cloud Tongyi Qianwen team, which supports multimodal input of text, images, and videos. It has 27 language processing capabilities and 32K long context understanding, and performs better than models of the same scale in areas such as code generation and mathematical reasoning. Adopting efficient group query attention (GQA) technology, adapted to various scenarios from mobile to server, and open sourced under Apache 2.0 protocol, it is commercially available for free.

4. Qwen1.5-14b (Bai et al., 2023): Qwen1.5-14b is a language model series including decoder language models of different model sizes. For each size, we release the base language model and the aligned chat model. It is based on the Transformer architecture with SwiGLU activation, attention QKV bias, group query attention, mixture of sliding window attention and full attention, etc.

C Limitations

Our study has several limitations. The reliance on LoRA’s framework somewhat limits its extensibility. Additionally, during the first stage of fine-tuning, the introduction of discrete noise does not specifically consider the distribution of noisy data, which, to a certain extent, increases the uncertainty. This presents a potential direction for future exploration, where more noise modeling techniques could be developed.

D Additional Experiments

D.1 Robustness Analysis under Varying Basic Noise Ratios

To investigate the robustness of LoPE when fine-tuning on datasets with higher inherent noise, we systematically evaluated basic noise ratios of 3.5%, 5%, 8%, 10%, 20%, and 30%. As shown in Table 8, although overall performance gradually declines as the noise level increases, it does not fully collapse, demonstrating that LoPE effectively delays performance degradation under high-noise conditions. This behavior can be attributed to several factors. First, the overall quality of fine-tuning datasets impacts final inference performance. When the noise reaches a certain level, it

Table 8: Performance under different noise ratios (transposed version).

Dataset / Noise Ratio	3.5%	5%	8%	10%	20%	30%	30% (HydraLoRA)
MMLU	44.59	44.42	44.15	43.81	42.35	40.78	37.96
PIQA+SIQA	63.31	62.66	61.86	61.27	60.14	59.50	56.83
GSM8K	15.31	13.72	11.83	9.25	8.04	7.60	7.12

blurs the boundary between clean data and noise, and fine-tuning becomes harmful, highlighting the importance of fine-tuning data quality. As a result, for tasks requiring high precision, such as mathematics datasets, the rate of accuracy decline actually slows at higher noise levels. This may be because mathematical pattern-related content in the fine-tuning dataset has already been corrupted, resulting in less impact on the model itself, and the model may no longer recognize that it is learning mathematics-related knowledge.

D.2 Specific accuracy for sub-task

The specific accuracy for each task can be found in Table 9, Table 10, Table 11, Table 12, Table 13, and Table 14.

Table 9: Performance across each specific task in NSET for LoPE and HydraLoRA (*Orig*).

Task	Model	
NSET	LoPE	HydraLoRA
College Biology	44.44	43.75
College Chemistry	32.00	30.00
College Computer Science	37.00	37.00
College Mathematics	38.00	34.00
College Physics	16.67	18.63
Electrical Engineering	42.76	45.52
Astronomy	40.79	42.76
Abstract Algebra	32.00	30.00
Machine Learning	38.39	36.61
Computer Security	58.00	58.00
High School Chemistry	35.96	32.51
Elementary Mathematics	29.89	27.51
High School Physics	28.48	29.14
High School Computer Science	37.00	37.00
High School Statistics	30.56	25.00
High School Biology	47.42	49.35
High School Mathematics	29.26	28.89
Conceptual Physics	44.26	42.13

Table 10: Performance across each specific task in SBH for LoPE and HydraLoRA(*Orig*).

Task	Model	
SBH	LoPE	HydraLoRA
Moral Scenarios	27.71	24.69
Professional Psychology	43.14	43.63
World Religions	66.67	66.08
Philosophy	59.49	58.84
Moral Disputes	49.13	51.16
High School Government and Politics	66.84	68.39
Sociology	61.19	64.18
High School Psychology	61.47	61.28

Table 11: Performance across each specific task in History for LoPE and HydraLoRA(*Orig*).

Task	Model	
History	LoPE	HydraLoRA
High European History	60.61	63.64
Prehistory	47.22	47.22
High School World History	62.45	59.49
High School US History	56.37	53.43

Table 12: Performance across each specific task in NSET for LoPE and HydraLoRA(*Nois*).

Task	Model	
NSET	LoPE	HydraLoRA
College Biology	45.14	43.75
College Chemistry	31.00	29.00
College Computer Science	38.00	33.00
College Mathematics	33.00	40.00
College Physics	17.65	15.69
Electrical Engineering	38.62	33.79
Astronomy	42.11	38.16
Abstract Algebra	32.00	34.00
Machine Learning	38.39	37.50
Computer Security	53.00	52.00
High School Chemistry	33.50	33.00
Elementary Mathematics	26.72	27.51
High School Physics	31.13	27.81
High School Computer Science	41.00	38.00
High School Statistics	21.30	23.61
High School Biology	46.13	47.10
High School Mathematics	28.52	29.63
Conceptual Physics	40.85	41.70

Table 13: Performance across each specific task in SBH for LoPE and HydraLoRA(*Nois*).

Task	Model	
SBH	LoPE	HydraLoRA
Moral Scenarios	26.03	25.92
Professional Psychology	44.28	44.12
World Religions	63.74	64.33
Philosophy	56.27	56.91
Moral Disputes	48.84	48.55
High School Government and Politics	65.28	64.25
Sociology	58.21	56.22
High School Psychology	60.37	58.53

Table 14: Performance across each specific task in History for LoPE and HydraLoRA(*Nois*).

Task	Model	
History	LoPE	HydraLoRA
High European History	63.61	61.21
Prehistory	49.38	47.53
High School World History	59.92	59.07
High School US History	54.41	55.39

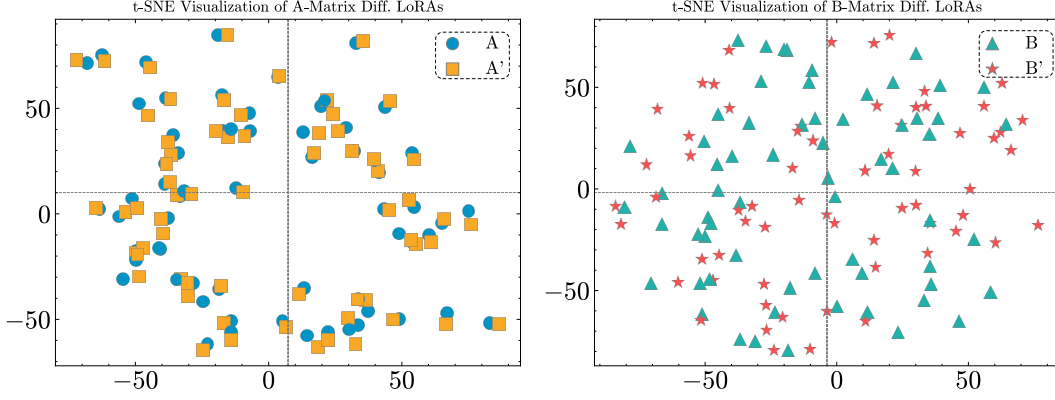


Figure 5: Comparison of the average parameter changes of matrices A and B during first-stage fine-tuning, with and without HyNoIse. Specifically, A and B correspond to the matrix A and poisoning expert B_D trained without HyNoIse, while A' and B' represent the respective matrix A and poisoning expert B_D trained with HyNoIse injected.

D.3 Matrix Parameter Visualization

We hypothesize that matrix A captures fundamental language-environment knowledge, which is both basic and universal, and is minimally affected by noise (a dataset with added noise can be considered analogous to a dataset from a new domain). To investigate this, we conducted Stage I training with and without HyNoIse data, observing the changes in matrix A parameters to assess whether harmful noise is absorbed into matrix A . Figure 5 shows that the average parameter change of matrix A , with or without HyNoIse, is much smaller than that of matrix B under the same operations. This finding suggests that matrix A primarily encodes universal contextual features present across all data, while the influence of noise on matrix A is substantially smaller than on matrix B .

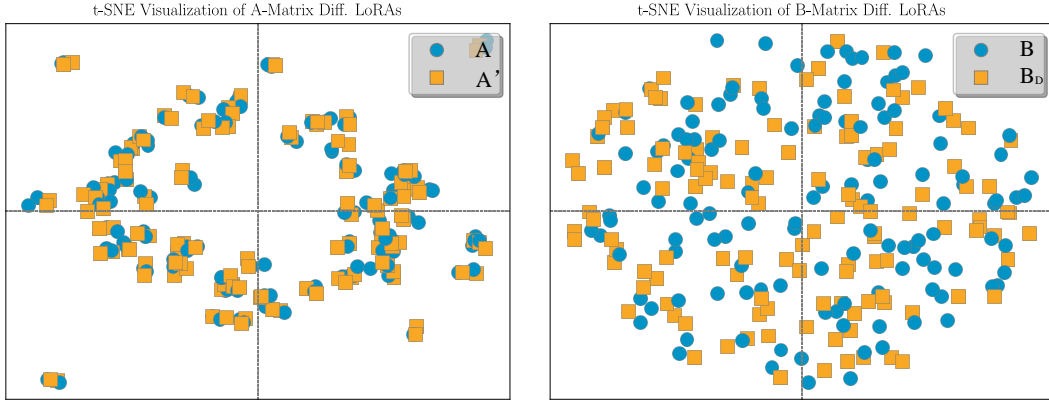


Figure 6: A denotes the parameter visualization of matrix A during the first-stage fine-tuning, while A' represents the updated matrix A in the second stage. B indicates a randomly selected normal expert in the second-stage fine-tuning, and B_D corresponds to the poisoning expert from the first stage. This visualization highlights the parameter evolution of clean and poisoned experts across different training phases.

We further visualize the parameter distributions of matrices A and B across the two-stage fine-tuning process. As shown in Figure 6, the position of matrix A in the visualization space changes very little throughout training. This observation that matrix A primarily encodes general, universal knowledge. In contrast, we observe significant positional shifts between a randomly selected normal expert matrix B and the poisoning expert matrix B_D across the two stages. We attribute this phenomenon to matrix

B capturing task-specific knowledge, while B_D absorbs noise-contaminated patterns, fundamentally differing from the clean representations encoded by other experts.

D.4 Evaluation against a Task-Specific Denoising Baseline

We have also included a dedicated comparison experiment against a task-specific denoising model to further assess LoPE’s generality in downstream applications. Specifically, we investigated the representative denoising method LeCoRE, which is designed for conversational search tasks. Following its evaluation protocol, we conducted experiments under comparable settings. The results are summarized in Table 15.

Table 15: Performance comparison between LoPE and the task-specific denoising model LeCoRE.

Method	MRR	NDCG@3	R@10	R@100
LeCoRE	51.1	48.5	73.9	89.7
LoPE	54.7	50.1	75.6	91.3