BA-LoRA: Bias-Alleviating Low-Rank Adaptation to Mitigate Catastrophic Inheritance in Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Parameter-efficient fine-tuning (PEFT) has become a de facto standard for adapting Large Language Models (LLMs). However, we identify a critical vulnerability within popular low-rank adaptation methods like LoRA: they can exacerbate "Catastrophic Inheritance"—the unchecked propagation of biases, noise, and data imbalances from pre-training. This phenomenon can degrade model robustness and fairness, undermining the benefits of efficient adaptation. To address this, we introduce Bias-Alleviating Low-Rank Adaptation (BA-LoRA). Our approach is founded on a principled decomposition of Catastrophic Inheritance into three core challenges: Knowledge Drift, Representation Collapse, and Overfitting to Noise. BA-LoRA systematically mitigates these issues by incorporating a trio of targeted regularizers—consistency, diversity, and SVD—designed to preserve core knowledge, enforce representational richness, and promote robust, low-rank output representations. We conduct comprehensive evaluations on a suite of natural language understanding (NLU) and generation (NLG) tasks using diverse, prominent open-source language models (e.g., LLaMA-2-7B and DeBERTa-v3-base). Our results show that BA-LoRA not only outperforms state-of-the-art LoRA variants in terms of performance and stability, but also demonstrates quantitatively superior robustness and bias mitigation on targeted evaluations. This confirms its ability to counteract the adverse effects of Catastrophic Inheritance.

1 Introduction

Large language models (LLMs) like GPT-4 (OpenAI, 2023) and LLaMA (Touvron et al., 2023) have redefined the state-of-the-art in natural language processing (NLP), largely due to their training on vast, web-scale corpora (Zhao et al., 2023; Chang et al., 2024). This strategy, while enabling unprecedented generalization (Gao et al., 2020; Penedo et al., 2023), comes at a cost: models inevitably inherit and internalize the biases, noise, and imbalances latent within these unfiltered datasets (Parashar et al., 2024; Liu & He, 2024; Chen et al., 2024b).

Recent research confirms that these inherited flaws can degrade model performance and persist even after fine-tuning, posing significant risks to fairness and safety (Qi et al., 2023; Bommasani et al., 2021; Mallen et al., 2022; Carlini et al., 2023). For example, noise within the training data can degrade model generalization (Chen et al., 2024a), while the long-tailed distribution of concepts can cause LLMs to overemphasize overrepresented topics (Zhu et al., 2024; Dong et al., 2023).

This phenomenon, termed "Catastrophic Inheritance" (Chen et al., 2024a), arises when models inherit such biases, noise, and imbalances from pre-training; we focus on how these inherited artifacts can be further amplified during downstream fine-tuning, and this has spurred investigations into mitigation strategies. While constructing less biased datasets and developing more robust model architectures are prominent approaches (Liu & He, 2024), this study explores an alternative: innovations in fine-tuning. Fine-tuning is a powerful method for enhancing task-specific performance and aligning models with user intent (Han et al., 2024; Ouyang et al., 2022). However, its computational demands are substantial; for instance, 16-bit fine-tuning of a Llama-65B model requires over 780 GB of GPU memory (Dettmers et al., 2024). To address these limitations, parameter-efficient fine-tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA) (Hu et al., 2021), have gained prominence.

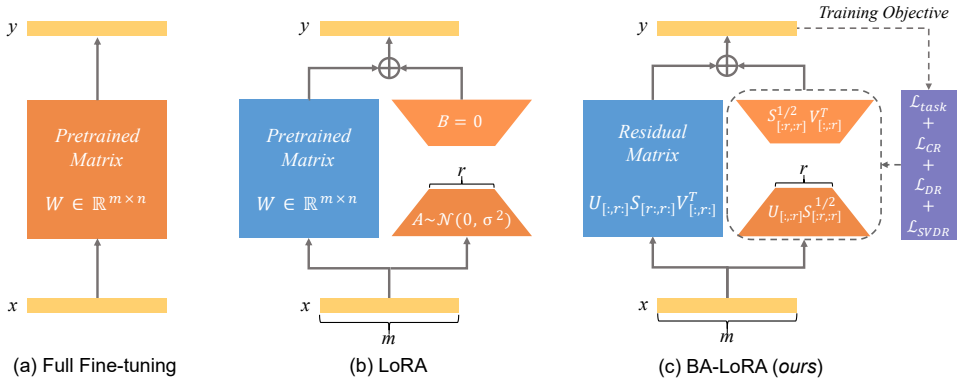


Figure 1: Comparison of three fine-tuning frameworks: (a) Full Fine-tuning, updating the entire matrix W; (b) LoRA, training a low-rank adapter for a frozen W; and (c) our proposed BA-LoRA. Blue and orange modules denote frozen and trainable parameters, respectively. Our method first initializes its adapter and residual matrix (W^{res}) from the SVD of W (PiSSA-style). It then augments the task loss ($\mathcal{L}_{\text{task}}$) with three regularization terms (purple module) designed to mitigate catastrophic inheritance by preserving knowledge, promoting diversity, and focusing on core data patterns.

LoRA enables efficient fine-tuning by approximating parameter updates using low-rank matrices. As illustrated in Figure 1 (a), Full Fine-tuning directly updates the entire weight matrix W. In contrast, LoRA (Figure 1 (b)) introduces a learnable low-rank adapter $\Delta W = AB$, where $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$ are trainable matrices with a rank $r \ll \min(m, n)$. Only A and B are updated, while the original weights W remain frozen. By initializing A with scaled random values and B to zero, LoRA ensures the adapter has no effect at the start of training. The forward pass is then computed as $Y = X(W + AB)$, significantly reducing computational costs (Hu et al., 2021).

While PEFT methods like LoRA offer remarkable efficiency, their constrained, low-rank updates introduce a critical vulnerability: they can exacerbate Catastrophic Inheritance when fine-tuning on noisy or imbalanced data without explicit regularization. By forcing all model adjustments through a low-dimensional bottleneck, these methods may lack the capacity to correct for inherited biases, instead amplifying spurious correlations from pre-training data. To bridge this gap, we argue that a more principled approach is needed. We first deconstruct Catastrophic Inheritance into three primary failure modes: **Knowledge Drift**, where the model unintentionally forgets robust pre-trained knowledge while learning new tasks (Kirkpatrick et al., 2017); **Representation Collapse**, where fine-tuning on imbalanced data causes output diversity to plummet (Bardes et al., 2021); and **Overfitting to Noise**, where the model learns spurious correlations from the training data that hinder generalization (Chen et al., 2019). This paper introduces Bias-Alleviating Low-Rank Adaptation (BA-LoRA), a novel method that systematically mitigates these issues. As depicted in Figure 1 (c), BA-LoRA builds upon the efficient PiSSA (Meng et al., 2024) initialization and incorporates a trio of regularizers: a consistency regularizer to combat Knowledge Drift, a diversity regularizer to prevent Representation Collapse, and an SVD regularizer to mitigate Overfitting to Noise. Recognizing the differences between NLU and NLG tasks, we tailor these strategies accordingly.

Our comprehensive evaluation establishes BA-LoRA's superior performance and deconstructs the sources of its effectiveness. BA-LoRA consistently outperforms leading LoRA variants across diverse benchmarks, including mathematical reasoning, coding, and conversational AI for NLG, as well as the GLUE benchmark (Wang et al., 2018) for NLU, using models such as LLaMA-2-7B (Touvron et al., 2023) and DeBERTa-v3-base (He et al., 2021). Crucially, we move beyond standard leaderboards to test our central hypothesis. An empirical comparison on models pre-trained with clean (RoBERTa (Liu et al., 2019)) versus noisier web-scale (T5 (Raffel et al., 2020)) data shows that BA-LoRA achieves larger gains on the latter, which is consistent with our hypothesis that BA-LoRA is particularly beneficial when mitigating the effects of inherited noise. This primary finding is supported by comprehensive ablation studies and qualitative visualizations that confirm the necessity of our three-pronged strategy. Together, these results not only demonstrate BA-LoRA's superiority but also validate our theoretical framework for understanding and mitigating this phenomenon.

2 METHOD

2.1 PRINCIPAL SINGULAR VALUES AND SINGULAR VECTORS ADAPTATION (PISSA)

As a variant of LoRA, PiSSA accelerates convergence by retaining the core LoRA architecture but changing the initialization. It leverages the principal components of the original weight matrix W to initialize the adapter matrices A and B, and stores the remaining components in a residual matrix $W^{\text{res}} \in \mathbb{R}^{m\times n}$. We write the SVD of $W \in \mathbb{R}^{m\times n}$ as $W = USV^T$, where U and V contain the left and right singular vectors and S is a diagonal matrix of singular values sorted in descending order. PiSSA partitions the singular components into principal $\{U_{[:,:r]}, S_{[:r,:r]}, V_{[:,:r]}\}$ and residual $\{U_{[:,r:]}, S_{[r:,r:]}, V_{[:,r:]}\}$ parts, where r is the user-specified adapter rank. The principal components are then used to initialize the low-rank adapter with $A \in \mathbb{R}^{m\times r}$ and $B \in \mathbb{R}^{r\times n}$:

$$A = U_{[:,:r]}\, S^{1/2}_{[:r,:r]} \in \mathbb{R}^{m\times r} \tag{1}$$

$$B = S^{1/2}_{[:r,:r]}\, V^T_{[:,:r]} \in \mathbb{R}^{r\times n} \tag{2}$$

The residual matrix W^{res} remains frozen during fine-tuning:

$$W^{\text{res}} = U_{[:,r:]}\, S_{[r:,r:]}\, V^T_{[:,r:]} \in \mathbb{R}^{m\times n} \tag{3}$$

PiSSA preserves the pre-trained model's full capacity at the start of fine-tuning by using $W = W^{\text{res}} + AB$. This approach prioritizes training the most influential parameters, thereby accelerating convergence from the start. Inheriting LoRA's benefits of reduced parameter count and deployment simplicity, PiSSA further leverages efficient SVD computations to expedite the training process. This concentration of updates also motivates the output-space regularizers introduced in BA-LoRA.

2.2 BIAS-ALLEVIATING LOW-RANK ADAPTATION (BA-LORA)

Catastrophic Inheritance refers to vulnerabilities from biases inherent in large-scale training data, particularly attribute bias and class imbalance, that degrade downstream performance, introduce unfair biases, and pose security risks. These effects manifest during fine-tuning as three distinct subproblems: **Knowledge Drift**, where the model unintentionally forgets or distorts robust pre-trained knowledge (Kirkpatrick et al., 2017); **Representation Collapse** (Bardes et al., 2021); and **Overfitting to Noise** (Chen et al., 2019). To address them, we propose BA-LoRA, a unified framework with three regularizers—consistency, diversity, and SVD—each aligned to one subproblem. Instead of constraining low-rank adapter weights, BA-LoRA regularizes the output space to directly shape functional behavior and mitigate bias, with tailored variants for NLU and NLG tasks.

2.2.1 REGULARIZATIONS FOR NLU TASKS

Consistency Regularization. To directly combat **Knowledge Drift**, we adopt a knowledge distillation approach based on standard practices (Hinton et al., 2015), using the Kullback-Leibler (KL) divergence between the temperature-scaled probability distributions. Let $\mathbf{Z}_P, \mathbf{Z}_F \in \mathbb{R}^{N\times D}$ be the batch output logits from the pre-trained and fine-tuned models respectively, where N is the batch size and D is the number of classes. The loss is defined as:

$$\mathcal{L}_{\text{CR\_NLU}} = T^2 \cdot \text{KL}(\text{softmax}(\mathbf{Z}_P/T) \,\|\, \text{softmax}(\mathbf{Z}_F/T)) \tag{4}$$

where T is a temperature parameter that softens the distributions. This objective encourages the fine-tuned model to mimic the nuanced decision-making process of the pre-trained model on examples where the teacher signal is reliable, preserving foundational knowledge. The T^2 scaling factor ensures gradient magnitudes are commensurate with standard cross-entropy loss.

Diversity Regularization. To counteract **Representation Collapse**, particularly on imbalanced datasets, we promote diversity in the model's predictions across a batch. Inspired by (Bardes et al., 2021), we regularize the batch-wise output logits to decorrelate the predictions for different classes.

Let $\mathbf{Z}_F \in \mathbb{R}^{N \times D}$ be the logit matrix for a batch. We first center the logits and then compute the $D \times D$ covariance matrix $C(\mathbf{Z}_F)$. The regularizer penalizes the off-diagonal elements of this matrix:

$$\mathcal{L}_{\text{DR\_NLU}} = \frac{1}{D} \sum_{i \neq j} [C(\mathbf{Z}_F)]_{i,j}^2 \tag{5}$$

where the covariance matrix is computed using its matrix form:

$$C(\mathbf{Z}_F) = \frac{1}{N-1} \mathbf{Z}_{\text{centered}}^T \mathbf{Z}_{\text{centered}}, \quad \text{where } \mathbf{Z}_{\text{centered}} = \mathbf{Z}_F - \bar{\mathbf{Z}}_F \tag{6}$$

Here, $\bar{\mathbf{Z}}_F$ is the matrix where each row is the mean logit vector computed over the batch. This loss discourages excessive correlation between the model's predictions for any two distinct classes across the batch, thus preventing the model from collapsing towards a few dominant classes.

Singular Value Decomposition Regularization. To mitigate **Overfitting to Noise** and encourage the model to learn robust features, we introduce a regularizer that encourages the spectral energy of the batch-wise output logit matrix to concentrate in its leading singular components. Inspired by the principle that dominant singular values capture the most salient data patterns (Chen et al., 2019), this regularizer incentivizes the model to form simpler, more coherent decision boundaries for samples within a batch, rather than fitting to high-frequency logit fluctuations that are poorly aligned with the task labels. On the fine-tuned logit matrix $\mathbf{Z}_F \in \mathbb{R}^{N \times D}$, we perform SVD and maximize the ratio of spectral energy concentrated in the top-k singular values:

$$\mathcal{L}_{\text{SVDR\_NLU}} = -\frac{\sum_{i=1}^{k} \sigma_i}{\sum_{j=1}^{\min(N,D)} \sigma_j} \tag{7}$$

where σ_i is the i-th largest singular value. The hyperparameter k controls the rank preference. In the NLU experiments, where the number of classes D is typically moderate, the computational cost of performing an exact SVD is minimal and poses no challenge to the training efficiency.

Overall Objective Function for NLU. The overall NLU objective is formulated as follows:

$$\mathcal{L}_{\text{NLU}} = \mathcal{L}_{\text{task\_NLU}} + \lambda_1 \mathcal{L}_{\text{CR\_NLU}} + \lambda_2 \mathcal{L}_{\text{DR\_NLU}} + \lambda_3 \mathcal{L}_{\text{SVDR\_NLU}} \tag{8}$$

where $\mathcal{L}_{\text{task\_NLU}}$ represents the standard cross-entropy loss function for the downstream task, and λ_1, λ_2, and λ_3 are weighting parameters to balance each regularization term's contribution.

2.2.2 Regularizations for NLG Tasks

Consistency Regularization. To combat **Knowledge Drift**, we employ temperature-controlled knowledge distillation (Hinton et al., 2015), using the Kullback-Leibler Divergence (KLD) between the output distributions of the fine-tuned (student) model, $\mathcal{P}_F$, and the pre-trained (teacher) model, $\mathcal{P}_P$. A temperature parameter, T, softens these distributions, compelling the student to learn the teacher's nuanced output, not just its top prediction, especially on tokens where the teacher distribution provides a meaningful and well-calibrated soft target. The loss is defined as:

$$\mathcal{L}_{\text{CR\_NLG}} = T^2 \cdot \frac{1}{M} \sum_{i=1}^{M} \text{KL}(\mathcal{P}_P(y_i \mid \mathbf{x}; T) \,\|\, \mathcal{P}_F(y_i \mid \mathbf{x}; T)) \tag{9}$$

where for an input sequence $\mathbf{x}$, y_i is the target token at position i, and $\mathcal{P}(y_i \mid \mathbf{x}; T) = \text{softmax}(\mathbf{z}_i / T)$ is the temperature-scaled conditional probability from the logit vector $\mathbf{z}_i$. The loss is averaged over all M valid (non-padded) tokens in the batch. The critical T^2 scaling factor maintains gradient magnitude consistency with standard distillation.

Diversity Regularization. To counteract **Representation Collapse** in generation, we address a fundamental challenge: naively maximizing the entropy of the entire vocabulary distribution conflicts with the task objective of producing coherent text (Gat et al., 2020). We resolve this with a novel *focused* entropy regularizer. Inspired by Top-K sampling, our method promotes diversity exclusively within the set of most plausible candidate tokens, denoted as $\mathcal{V}_{\text{top-k}}$. For each token, we define the loss as the negative entropy computed solely within this restricted set:

$$\mathcal{L}_{\text{DR\_NLG}} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{j \in \mathcal{V}_{\text{top-k}}^{(i)}} P'_F(x_j | \mathbf{h}_i) \log P'_F(x_j | \mathbf{h}_i) \tag{10}$$

where $P'_F(x_j|\mathbf{h}_i)$ is the re-normalized probability from the fine-tuned model for token x_j within the set $\mathcal{V}^{(i)}_{\text{top-k}}$ for the i-th valid token, given the corresponding final hidden state $\mathbf{h}_i$.

Singular Value Decomposition Regularization. To mitigate **Overfitting to Noise**, we regularize the structure of the batch-wise output logit matrix. Building on the principle that dominant singular values capture salient data patterns (Chen et al., 2019), we encourage a low-rank structure. For tractability with large vocabularies, we use randomized SVD (Halko et al., 2011) and normalize by the efficient Frobenius norm to avoid expensive full-spectrum computation. We thus define the loss as the negative ratio of the sum of the top-k singular values to the Frobenius norm:

$$\mathcal{L}_{\text{SVDR\_NLG}} = -\frac{\sum_{i=1}^{k} \tilde{\sigma}_i}{\|\mathbf{Z}_{\text{valid}}\|_F} \tag{11}$$

Here, $\tilde{\sigma}_i$ is the i-th largest approximated singular value of the valid logit matrix $\mathbf{Z}_{\text{valid}} \in \mathbb{R}^{M \times |\mathcal{V}|}$, where $|\mathcal{V}|$ is the vocabulary size, and $\|\cdot\|_F$ denotes the Frobenius norm.

Overall Objective Function for NLG. Integrating these components, the final objective function for NLG tasks is a weighted sum of the task loss and our three regularization terms:

$$\mathcal{L}_{\text{NLG}} = \mathcal{L}_{\text{task\_NLG}} + \lambda_1 \mathcal{L}_{\text{CR\_NLG}} + \lambda_2 \mathcal{L}_{\text{DR\_NLG}} + \lambda_3 \mathcal{L}_{\text{SVDR\_NLG}} \tag{12}$$

where $\mathcal{L}_{\text{task\_NLG}}$ is the standard causal language modeling loss. Our experiments revealed the Minimal Intervention Principle: robust fine-tuning is best achieved by applying regularizers with minimal weights to gently guide the model. A detailed sensitivity analysis is provided in Appendix C.2.

3 Experiments

3.1 Implementation Details

Our experimental setup is broadly aligned with recent PEFT studies (Meng et al., 2024). For NLG tasks on LLaMA-2-7B, we use the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 2×10^{-5}, a cosine schedule (0.03 warmup ratio), and no weight decay. We set `lora_dropout` to 0, use BFloat16 precision, a LoRA rank (r) and alpha (α) of 128, and an effective batch size of 32. The key regularization weights for our method are set to $\lambda_1 = 0.025, \lambda_2 = 0.005, \lambda_3 = 0.005$, with an SVD rank of $k = 10$. For NLU tasks on the GLUE benchmark, learning rates, batch sizes, and other core hyperparameters are task-specific to strictly align with our baseline, as detailed in the appendix. For our method in the NLU setting, we use no weight decay and set regularization hyperparameters to $\lambda_1 = 0.15, \lambda_2 = 0.03, \lambda_3 = 0.03$ with an SVD rank $k = 5$. For each backbone, regularization weights are selected via coarse grid search on a held-out validation split and then kept fixed for that setting (see Appendices C.2 and D.1 for details). All experiments were conducted on NVIDIA A40 GPUs and averaged over three random seeds (42, 1024, 2024). Full and detailed hyperparameter configurations for all models and tasks are available in Appendix B.

3.2 Results and Analysis

Table 1: Performance comparison on NLG tasks. We compare our method (BA-LoRA) against popular fine-tuning baselines, including Full Fine-tuning and various state-of-the-art parameter-efficient techniques. The best results in each column are highlighted in **bold**.

Methods	GSM8K	MATH	HumanEval	MBPP	MT-Bench	Avg
Full FT	$48.9_{\pm 0.49}$	$7.48_{\pm 0.22}$	$20.52_{\pm 0.29}$	$23.64_{\pm 0.38}$	$4.85_{\pm 0.09}$	21.08
LoRA	$42.68_{\pm 0.54}$	$5.92_{\pm 0.15}$	$16.80_{\pm 0.38}$	$21.51_{\pm 0.43}$	$4.60_{\pm 0.14}$	18.30
AdaLoRA	$41.95_{\pm 0.90}$	$6.24_{\pm 0.38}$	$18.10_{\pm 0.46}$	$20.19_{\pm 0.71}$	$4.79_{\pm 0.18}$	18.25
DoRA	$41.77_{\pm 0.74}$	$6.20_{\pm 0.48}$	$16.86_{\pm 0.54}$	$21.60_{\pm 0.49}$	$4.48_{\pm 0.14}$	18.18
MiLoRA	$43.09_{\pm 1.16}$	$6.31_{\pm 0.39}$	$17.55_{\pm 0.24}$	$20.22_{\pm 0.37}$	$4.50_{\pm 0.17}$	18.33
LoRA+	$47.84_{\pm 0.39}$	$7.21_{\pm 0.49}$	$20.07_{\pm 0.38}$	$23.69_{\pm 0.29}$	$5.11_{\pm 0.06}$	20.78
LoRA-FA	$40.25_{\pm 0.46}$	$5.66_{\pm 0.47}$	$15.91_{\pm 0.41}$	$20.01_{\pm 0.32}$	$4.67_{\pm 0.12}$	17.30
LoRA-GA	$50.47_{\pm 0.98}$	$7.13_{\pm 0.44}$	$19.44_{\pm 0.45}$	$23.05_{\pm 0.40}$	$5.04_{\pm 0.10}$	21.03
PiSSA	$51.48_{\pm 0.34}$	$7.60_{\pm 0.18}$	$19.48_{\pm 0.45}$	$23.84_{\pm 0.46}$	$4.92_{\pm 0.07}$	21.46
CorDA	$53.90_{\pm 0.56}$	$8.52_{\pm 0.27}$	$21.03_{\pm 0.37}$	$24.15_{\pm 0.44}$	$5.15_{\pm 0.09}$	22.55
CorDA++	$55.03_{\pm 0.52}$	$8.95_{\pm 0.37}$	$21.76_{\pm 0.39}$	$24.74_{\pm 0.47}$	$\mathbf{5.64}_{\pm 0.12}$	23.22
BA-LoRA	$\mathbf{55.86}_{\pm 0.35}$	$\mathbf{9.47}_{\pm 0.52}$	$\mathbf{23.58}_{\pm 0.25}$	$\mathbf{36.86}_{\pm 0.31}$	$5.11_{\pm 0.05}$	**25.90**

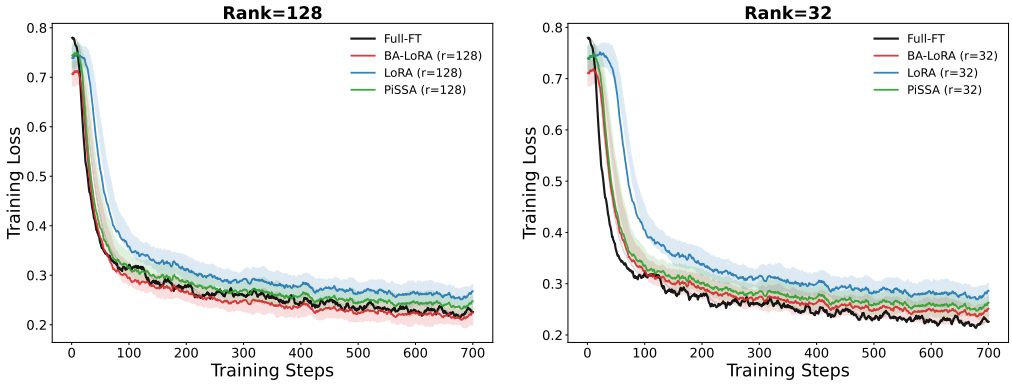


Figure 2: Training task loss of Full Fine-Tuning (Full FT), LoRA, PiSSA, and BA-LoRA on MetaMath: (left) rank 128 and (right) rank 32. All curves are smoothed for visual clarity.

3.2.1 Performance on NLG and NLU Tasks

To evaluate BA-LoRA on NLG tasks, we conduct a fair comparison against strong baselines (Table 1), sourcing their scores from original publications with comparable setups (see Appendix B.3). These baselines include Full Fine-tuning, LoRA (Hu et al., 2021), AdaLoRA (Zhang et al., 2023b), DoRA (Liu et al., 2024), MiLoRA (Wang et al., 2024a), LoRA+ (Hayou et al., 2024), LoRA-FA (Zhang et al., 2023a), LoRA-GA (Wang et al., 2024b), PiSSA (Meng et al., 2024), CorDA (Yang et al., 2024), and CorDA++ (Yang et al., 2025). We fine-tuned LLaMA-2-7B (Touvron et al., 2023) on MetaMathQA (Yu et al., 2023) and assessed mathematical problem-solving capabilities using the GSM8K (Cobbe et al., 2021) and MATH (Yu et al., 2023) validation sets, reporting Accuracy. Similarly, models were fine-tuned on CodeFeedback (Zheng et al., 2024) and evaluated for coding via HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021), with PASS@1 metrics reported. To assess conversational abilities, models were trained on WizardLM-Evol-Instruct (Xu et al., 2024) and evaluated on MT-Bench (Zheng et al., 2024), with response quality judged by GPT-4 and first-turn scores reported. All experiments utilized 100K data points and a single epoch for efficiency.

As shown in Table 1, BA-LoRA establishes a new state-of-the-art on LLaMA-2-7B, outperforming the strongest baselines. Specifically, compared to the highly competitive CorDA++, BA-LoRA further enhances performance on the reasoning task GSM8K by 0.83 points and the coding task HumanEval by 1.82 points. While CorDA++ maintains an edge on MT-Bench, BA-LoRA's substantial gains on other benchmarks lead to a superior average score, achieving a 2.68-point uplift over CorDA++, with the largest margin on MBPP, a small natural language–to–code benchmark with limited test coverage and susceptibility to overfitting and spurious patterns, where BA-LoRA's output-space regularization is helpful under noisy supervision. This performance improvement is further corroborated by the model's optimization dynamics. As illustrated in Figure 2, BA-LoRA also demonstrates superior training efficiency on the MetaMath dataset. Across both high ($r = 128$) and low ($r = 32$) ranks, our method achieves a lower final training task loss than LoRA and PiSSA, reaching levels comparable to Full FT, which we attribute to our principled regularization scheme guiding the optimization toward a more favorable solution space.An extensive comparison across diverse model families and scales, including dense and MoE models from 7B to LLaMA-3-70B, is provided in Appendix C.3 (Figure 5).

To assess BA-LoRA on NLU tasks, we experimented on the GLUE benchmark (Wang et al., 2018), which includes two single-sentence classification tasks (CoLA, SST), five paired-text classification tasks (MNLI, RTE, QQP, MRPC, QNLI), and one text similarity prediction task (STS-B). The evaluation metrics comprise the overall matched and mismatched accuracy for MNLI, the Matthews correlation coefficient for CoLA, the Pearson correlation coefficient for STS-B, and the accuracy for the remaining tasks. We used the DeBERTa-v3-base model (He et al., 2021) and compared BA-LoRA against eight strong baseline methods, including Full Fine-Tuning (Full FT), BitFit (Zaken et al., 2021), HAdapter (Houlsby et al., 2019), PAdapter (Pfeiffer et al., 2020), LoRA (Hu et al., 2021), DoRA (Liu et al., 2024), AdaLoRA (Zhang et al., 2023b), and PiSSA (Meng et al., 2024).

Table 2 presents the results of DeBERTa-v3-base across eight NLU tasks, demonstrating the strong overall performance of BA-LoRA. It surpasses all parameter-efficient fine-tuning (PEFT) baselines

on every task and achieves the highest average score. On average, BA-LoRA outperforms PiSSA and LoRA by 1.20 and 2.11 points, respectively. The consistent, broad-based improvements across this diverse suite of both NLG and NLU tasks provide strong evidence that our principled, three-pronged strategy not only mitigates the fundamental failure modes associated with Catastrophic Inheritance but also offers a practical route to state-of-the-art performance with parameter-efficient adaptation.

Table 2: Performance Comparison on NLU Benchmarks. We compare BA-LoRA with various PEFT baselines on the DeBERTa-v3-base model. The best result in each column is highlighted in bold.

Methods	#Params	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg
Full FT	184M	$90.34_{\pm 0.18}$	$\mathbf{96.33}_{\pm 0.11}$	$89.95_{\pm 1.07}$	$71.43_{\pm 0.72}$	$94.24_{\pm 0.10}$	$92.11_{\pm 0.28}$	$83.75_{\pm 1.81}$	$91.04_{\pm 0.48}$	88.86
BitFit	0.1M	$89.54_{\pm 0.29}$	$94.68_{\pm 0.11}$	$87.95_{\pm 1.33}$	$67.31_{\pm 0.49}$	$92.45_{\pm 0.17}$	$88.72_{\pm 0.45}$	$79.12_{\pm 0.39}$	$91.63_{\pm 0.37}$	86.43
HAdapter	1.22M	$90.23_{\pm 0.07}$	$95.38_{\pm 0.06}$	$89.97_{\pm 0.27}$	$68.73_{\pm 0.27}$	$94.31_{\pm 0.29}$	$91.99_{\pm 0.28}$	$84.76_{\pm 0.39}$	$91.58_{\pm 0.13}$	88.37
PAdapter	1.18M	$90.42_{\pm 0.36}$	$95.49_{\pm 0.10}$	$89.71_{\pm 0.35}$	$69.04_{\pm 0.10}$	$94.38_{\pm 0.26}$	$92.15_{\pm 0.43}$	$85.53_{\pm 0.18}$	$91.69_{\pm 0.13}$	88.55
LoRA	1.33M	$90.71_{\pm 0.16}$	$94.79_{\pm 0.16}$	$89.85_{\pm 0.21}$	$70.05_{\pm 0.34}$	$93.94_{\pm 0.09}$	$92.07_{\pm 0.48}$	$85.43_{\pm 0.09}$	$91.67_{\pm 0.29}$	88.56
DoRA	1.27M	$90.48_{\pm 0.10}$	$95.85_{\pm 0.08}$	$91.04_{\pm 0.15}$	$71.03_{\pm 0.18}$	$94.21_{\pm 0.37}$	$92.34_{\pm 0.16}$	$86.19_{\pm 0.25}$	$91.92_{\pm 0.38}$	89.13
AdaLoRA	1.27M	$90.87_{\pm 0.08}$	$96.18_{\pm 0.43}$	$90.81_{\pm 0.40}$	$71.64_{\pm 0.12}$	$94.68_{\pm 0.46}$	$92.37_{\pm 0.35}$	$87.78_{\pm 0.36}$	$91.97_{\pm 0.43}$	89.53
PiSSA	1.33M	$90.47_{\pm 0.44}$	$95.81_{\pm 0.45}$	$91.48_{\pm 0.49}$	$72.27_{\pm 0.29}$	$94.41_{\pm 0.41}$	$92.21_{\pm 0.26}$	$87.14_{\pm 0.08}$	$91.93_{\pm 0.25}$	89.47
BA-LoRA	1.33M	$\mathbf{91.26}_{\pm 0.49}$	$96.25_{\pm 0.09}$	$\mathbf{92.11}_{\pm 0.55}$	$\mathbf{75.46}_{\pm 0.62}$	$\mathbf{95.35}_{\pm 0.14}$	$\mathbf{93.63}_{\pm 0.52}$	$\mathbf{88.58}_{\pm 0.73}$	$\mathbf{92.71}_{\pm 0.38}$	**90.67**

3.2.2 Mitigating the Effects of Noisy Pre-training Data

Given that large-scale pre-training corpora from web crawls are inherently noisy (Gao et al., 2020; Dodge et al., 2021), a critical challenge is ensuring that fine-tuning enhances the core signal rather than inherited noise. To investigate BA-LoRA's ability to address this, we conduct a controlled study on models pre-trained on corpora of distinct quality. Our testbeds are RoBERTa-base (Liu et al., 2019), pre-trained on a high-quality, curated corpus, and T5-base (Raffel et al., 2020), pre-trained on the noisier, large-scale C4 web corpus.[1] While these models differ in architecture, their distinct pre-training corpora provide an ideal but not fully controlled testbed for evaluating robustness against inherited noise. We evaluate on a representative subset of the GLUE benchmark.

As detailed in Table 3, BA-LoRA achieves the best average performance against strong PEFT baselines. The central finding is that the advantage of BA-LoRA is significantly more pronounced on the model pre-trained on noisier data. While BA-LoRA establishes a solid 1.11-point average improvement over the strongest baseline (PiSSA) on the cleanly-trained RoBERTa-base (86.34 vs. 85.23), this performance gain nearly triples to a substantial 3.26 points on the T5-base (87.97 vs. 84.71). The pronounced disparity in improvement margin ($\Delta_{\text{T5}} = 3.26$ vs. $\Delta_{\text{RoBERTa}} = 1.11$) is consistent with our hypothesis that BA-LoRA is particularly beneficial for models pre-trained on noisier web corpora, though it does not isolate architectural factors.

3.2.3 Mitigating Representational Bias from Data Imbalance

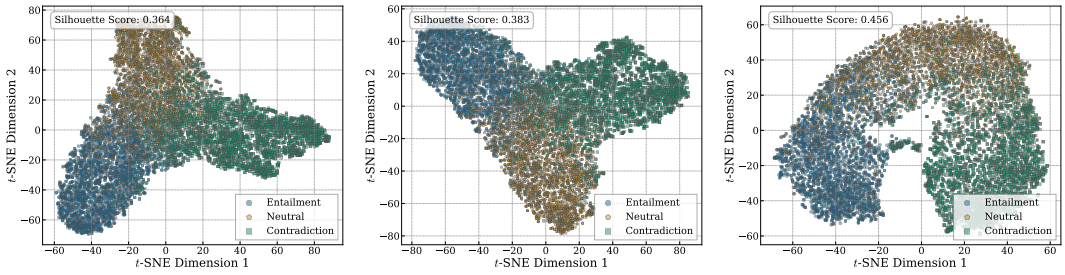
This experiment qualitatively investigates BA-LoRA's capacity to counteract the representational degradation caused by data imbalance, which is one way Catastrophic Inheritance can manifest during downstream fine-tuning. We visualize the final hidden-layer feature representations from RoBERTa-base fine-tuned on the MNLI task using t-SNE (Van der Maaten & Hinton, 2008). As shown in Figure 3, we compare feature manifolds learned on the standard balanced dataset against those from a deliberately imbalanced version—constructed by subsampling the training data to a 100:10:1 ratio for the 'Entailment', 'Neutral', and 'Contradiction' classes. This controlled comparison is designed to simulate the challenge of learning from highly skewed data distributions on top of a pre-trained model, where Catastrophic Inheritance–style failures can arise.

The visualization contrasts the methods' resilience to data imbalance. While representations from baseline LoRA and PiSSA suffer degradation and class overlap (Figure 3d,e), BA-LoRA maintains a well-separated manifold (Figure 3f; see Table 12 in Appendix C.6 for quantitative verification). This visualization is consistent with the effectiveness of our diversity regularizer ($\mathcal{L}_{\text{DR}}$) in preventing feature degradation from skewed data distributions. This effect is reinforced by the consistency ($\mathcal{L}_{\text{CR}}$) and SVD ($\mathcal{L}_{\text{SVDR}}$) regularizers, which together help the representations remain distinct and robust.

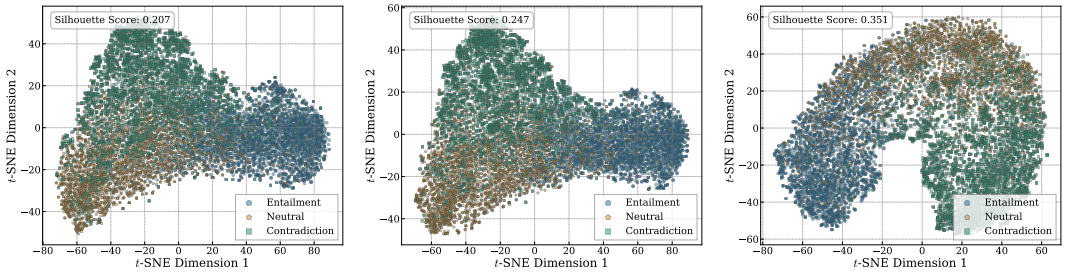
[1]The C4 corpus (Colossal Clean Crawled Corpus) is derived from the broad Common Crawl web scrape via heuristic filtering. While RoBERTa's ∼160GB dataset also includes web text, it is a curated mixture containing high-purity sources like BooksCorpus and English Wikipedia. In contrast, C4 (∼750GB) is a larger, more homogeneous corpus drawn from a rawer source, making it a more representative testbed for web-scale noise.

Table 3: Performance comparison of our method (BA-LoRA) against PEFT baselines (LoRA, PiSSA) on RoBERTa-base and T5-base. Models are evaluated on a subset of the GLUE benchmark. The best result for each model is in bold.

Model	Methods	MNLI	SST-2	CoLA	QNLI	MRPC	Avg
RoBERTa-base	LoRA	$85.63_{\pm 0.01}$	$94.03_{\pm 0.02}$	$62.40_{\pm 0.71}$	$91.37_{\pm 0.97}$	$87.98_{\pm 0.23}$	84.28
	PiSSA	$85.72_{\pm 0.40}$	$93.64_{\pm 0.13}$	$67.28_{\pm 0.59}$	$91.40_{\pm 0.54}$	$88.11_{\pm 0.24}$	85.23
	BA-LoRA	$\mathbf{86.59}_{\pm 0.58}$	$\mathbf{94.83}_{\pm 0.45}$	$\mathbf{67.91}_{\pm 0.21}$	$\mathbf{92.28}_{\pm 0.37}$	$\mathbf{90.07}_{\pm 0.32}$	**86.34**
T5-base	LoRA	$85.30_{\pm 0.04}$	$94.04_{\pm 0.11}$	$69.35_{\pm 0.05}$	$92.96_{\pm 0.09}$	$68.38_{\pm 0.01}$	82.08
	PiSSA	$85.75_{\pm 0.07}$	$94.07_{\pm 0.06}$	$74.27_{\pm 0.39}$	$93.15_{\pm 0.14}$	$76.31_{\pm 0.51}$	84.71
	BA-LoRA	$\mathbf{86.91}_{\pm 0.48}$	$\mathbf{95.20}_{\pm 0.29}$	$\mathbf{80.19}_{\pm 1.03}$	$\mathbf{94.12}_{\pm 0.32}$	$\mathbf{83.43}_{\pm 0.71}$	**87.97**



(a) LoRA, Balanced (b) PiSSA, Balanced (c) BA-LoRA, Balanced



(d) LoRA, Imbalanced (e) PiSSA, Imbalanced (f) BA-LoRA, Imbalanced

Figure 3: t-SNE visualizations of features from RoBERTa-base fine-tuned with LoRA, PiSSA, and BA-LoRA on the MNLI task under balanced (top) and imbalanced (bottom) settings.

3.2.4 Ablation Study

Our ablation study (Table 4) empirically supports our principled deconstruction of Catastrophic Inheritance. On NLG tasks with the LLaMA-2-7B model, we observe a consistent pattern: each regularizer yields a positive contribution over the baseline ('w/o Reg'). Specifically, gains from the consistency regularizer ($\mathcal{L}_{\text{CR}}$) support its role in combating Knowledge Drift by preserving foundational knowledge. Similarly, improvements from the diversity regularizer ($\mathcal{L}_{\text{DR}}$) highlight the importance of preventing Representation Collapse, and the significant contribution from the SVD regularizer ($\mathcal{L}_{\text{SVDR}}$) confirms the benefit of mitigating Overfitting to Noise.

This trend is mirrored in NLU tasks, where the DeBERTa-v3-base model also shows a clear uplift with each regularizer over the baseline. The full BA-LoRA model, which synergistically combines all three components, consistently achieves the highest performance across all evaluated settings. In summary, these results provide strong evidence that Knowledge Drift, Representation Collapse, and Overfitting to Noise are important and complementary failure modes in fine-tuning. Consequently, our integrated, multi-pronged solution yields strong generalization and robustness across both NLU and NLG domains. The selection of our regularization coefficients (λ_1, λ_2, λ_3) is validated by a detailed sensitivity analysis in Figure 4, and Table 10 further shows that the same regularization framework consistently improves multiple LoRA-style methods (LoRA, DoRA, PiSSA).

3.2.5 Computational Cost Analysis

Table 4: Ablation study of BA-LoRA regularizations on GSM8K, MATH, and NLU tasks. Results on GSM8K and MATH are from LLaMA-2-7B, while the NLU task refers to the average GLUE score from DeBERTa-v3-base. "Baseline" (PiSSA) is fine-tuned without our proposed regularizations. $\mathcal{L}_{\text{CR}}$, $\mathcal{L}_{\text{DR}}$, and $\mathcal{L}_{\text{SVDR}}$ denote adding only a single corresponding regularization to the baseline. "BA-LoRA (Full)" is the full model using all regularizations.

Configuration	GSM8K	MATH	Average of GLUE
Baseline (PiSSA)	$51.48_{\pm 0.34}$	$7.60_{\pm 0.18}$	89.47
$\mathcal{L}_{\text{CR}}$	$54.25_{\pm 0.59}$	$9.15_{\pm 0.25}$	90.18
$\mathcal{L}_{\text{DR}}$	$53.60_{\pm 0.46}$	$8.95_{\pm 0.18}$	89.85
$\mathcal{L}_{\text{SVDR}}$	$52.95_{\pm 0.55}$	$8.70_{\pm 0.22}$	89.71
BA-LoRA (Full)	$\mathbf{55.86_{\pm 0.35}}$	$\mathbf{9.47_{\pm 0.52}}$	**90.67**

To quantitatively evaluate the computational efficiency and performance of our method, we conducted a comparative experiment on two A40 (48GB) GPUs using DeepSpeed (Rasley et al., 2020) ZeRO-2 optimization. We fine-tuned the LLaMA-2-7B model on the first 100,000 entries of the MetaMathQA dataset. This experiment benchmarked four methods: full fine-tuning (Full FT), LoRA, PiSSA, and our proposed BA-LoRA. For each method, we measured the peak GPU memory consumption and the total training time to assess computational cost. Model performance was subsequently evaluated on the GSM8K benchmark.

Table 5: Computational Cost and Performance Comparison. Costs are measured on two A40 GPUs for fine-tuning LLaMA-2-7B.

Method	Memory Cost	Training Time	GSM8K
Full FT	>96 GB	>24h	$48.9_{\pm 0.49}$
LoRA	66.32 GB	4h 31min	$42.68_{\pm 0.54}$
PiSSA	66.59 GB	4h 17min	$51.48_{\pm 0.34}$
BA-LoRA	77.34 GB	4h 48min	$\mathbf{55.86}_{\pm 0.35}$

The results in Table 5 quantify the performance-cost trade-offs of various methods. BA-LoRA sets a new state-of-the-art with a GSM8K score of 55.86, significantly outperforming all baselines. This substantial performance gain is achieved with a modest overhead compared to PiSSA (+10.75 GB memory, +31 min training), highlighting a compelling performance-cost balance.

4 Related Work

Our work bridges two critical research areas. **In Parameter-Efficient Fine-Tuning (PEFT)**, our method builds upon Low-Rank Adaptation (LoRA) (Hu et al., 2021). Numerous LoRA variants have focused on enhancing performance and efficiency, such as QLoRA (Dettmers et al., 2024) and PiSSA (Meng et al., 2024). Crucially, while recent work has identified LoRA's low-rank update as a potential bottleneck that can interfere with pre-trained knowledge (Zhang et al., 2023a), the systematic mitigation of inherited biases remains a significant gap (see Appendix A.4). **In Bias Mitigation**, addressing biases from web-scale corpora is a foundational concern (Bender et al., 2021). While a rich literature exists on data filtering (Dodge et al., 2021) and algorithmic adjustments for full fine-tuning—such as representation debiasing (Ravfogel et al., 2020) and decoding strategies (Sheng et al., 2019) (see (Gallegos et al., 2024) for a survey)—these are not directly applicable to PEFT. Although recent analyses have begun to probe fairness issues within PEFT (Ding et al., 2024), BA-LoRA is, to our knowledge, the first to propose a concrete, multi-faceted algorithmic framework. It moves beyond analysis to systematically mitigate the broader problem of Catastrophic Inheritance by integrating a principled regularization scheme directly into the LoRA-based process.

5 Conclusion

This paper introduces BA-LoRA, a novel parameter-efficient fine-tuning framework aimed at mitigating Catastrophic Inheritance. Our core contribution is a principled approach that decomposes this challenge into three sub-problems—Knowledge Drift, Representation Collapse, and Overfitting to Noise—and addresses them with three targeted regularizers. Extensive experiments across NLG and NLU benchmarks support our integrated strategy, which achieves strong performance and improves robustness to inherited data biases relative to standard LoRA variants. By addressing Catastrophic Inheritance explicitly within the LoRA framework, BA-LoRA offers a more reliable pathway to adapt pre-trained models to downstream tasks where robustness and fairness are important.

REFERENCES

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

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

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.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1536–1546. IEEE, 2021.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*, 2023.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. Quality at a glance: An audit of web-crawled multilingual datasets. *arXiv e-prints*, pp. arXiv–2103, 2021.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

Hao Chen, Jindong Wang, Ankit Shah, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, and Bhiksha Raj. Understanding and mitigating the label noise in pre-training on downstream tasks. *arXiv preprint arXiv:2309.17002*, 2023.

Hao Chen, Bhiksha Raj, Xing Xie, and Jindong Wang. On catastrophic inheritance of large foundation models. *arXiv preprint arXiv:2402.01909*, 2024a.

Hao Chen, Jindong Wang, Ankit Shah, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, and Bhiksha Raj. Understanding and mitigating the label noise in pre-training on downstream tasks, 2024b.

Hao Chen, Zihan Wang, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, Bhiksha Raj, and Jindong Wang. Impact of noisy supervision in foundation model learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Haolin Chen and Philip N Garner. Bayesian parameter-efficient fine-tuning for overcoming catastrophic forgetting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pp. 1081–1090. PMLR, 2019.

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

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhoujie Ding, Ken Ziyu Liu, Pura Peetathawatchai, Berivan Isik, and Sanmi Koyejo. On fairness of low-rank adaptation of large models, 2024. URL https://arxiv.org/abs/2405.17512.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*, 2021.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*, 2023.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. What's in my big data? *arXiv preprint arXiv:2310.20707*, 2023.

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7382–7392, 2024.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179, 2024.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Itai Gat, Idan Schwartz, Alexander Schwing, and Tamir Hazan. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. *Advances in Neural Information Processing Systems*, 33:3197–3208, 2020.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288, 2011.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Jiale Kang and Qingyu Yin. Balancing lora performance and efficiency with simple shard sharing, 2025. URL https://arxiv.org/abs/2409.15371.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? *arXiv preprint arXiv:2403.08632*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models, 2024.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research (JAIR)*, 70:1373–1411, 2021.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Shubham Parashar, Zhiqiu Lin, Tian Liu, Xiangjue Dong, Yanan Li, Deva Ramanan, James Caverlee, and Shu Kong. The neglected tails in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12988–12997, 2024.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

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.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *SIGKDD*, pp. 3505–3506, 2020.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667*, 2020.

Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. Data contamination through the lens of time. *arXiv preprint arXiv:2310.10628*, 2023.

Rylan Schaeffer. Pretraining on the test set is all you need. *arXiv preprint arXiv:2309.08632*, 2023.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*, 2019.

James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 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.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 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.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. *arXiv preprint arXiv:2406.09044*, 2024a.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931, 2024b.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. Corda: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37: 71768–71791, 2024.

Yibo Yang, Sihao Liu, Chuan Rao, Bang An, Tiancheng Shen, Philip HS Torr, Ming-Hsuan Yang, and Bernard Ghanem. Dynamic context-oriented decomposition for task-aware low-rank adaptation with less forgetting and faster convergence. *arXiv preprint arXiv:2506.13187*, 2025.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023a.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023b.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Beier Zhu, Kaihua Tang, Qianru Sun, and Hanwang Zhang. Generalized logit adjustment: Calibrating fine-tuned models by removing label bias in foundation models. *Advances in Neural Information Processing Systems*, 36, 2024.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Appendix

CONTENTS

A Background **16**

A.1 Challenges of Bias and Noise in Pre-training Data 16

A.2 Mitigating Bias through Parameter-Efficient Fine-Tuning 17

A.3 Typologies of Noise in Pre-training Data 17

A.4 Extended Related Work 17

B Experimental Setup **18**

B.1 Models 18

B.2 Tasks, Datasets, and Metrics 18

B.3 Implementation and Training Details 19

B.4 Hyperparameter Settings 20

C More Experiments **21**

C.1 Generality of the Regularization Framework 21

C.2 Hyperparameter Sensitivity Analysis 21

C.3 Analysis Across Diverse Model Architectures and Scales 22

C.4 Performance Analysis Across Different Ranks 22

C.5 Comparison of Regularization Objectives for NLU 23

C.6 Quantitative Analysis of Cluster Quality and Minority Representation 23

D More Discussions **24**

D.1 Practical Hyperparameter Guidelines 24

D.2 Additional Discussion on RoBERTa vs. T5 24

D.3 Discussion on the Choice of Regularization Targets 25

D.4 Conceptual Foundations and Synergy of Regularizers 25

D.5 On Applying Representation Learning Principles during Fine-Tuning 26

D.6 Limitations and Future Work 26

D.7 Ethics Statement 26

D.8 Reproducibility 26

A BACKGROUND

A.1 CHALLENGES OF BIAS AND NOISE IN PRE-TRAINING DATA

Bias and noise within pre-training datasets present significant hurdles in constructing dependable machine-learning models. Mislabeled data and imbalanced distributions can lead to models that not only underperform on downstream tasks but also reinforce existing biases (Barocas & Selbst, 2016; Gallegos et al., 2024). This issue is especially problematic in large-scale datasets where

manual curation is impractical, and reliance on automated data collection may introduce various inaccuracies (Northcutt et al., 2021; Birhane & Prabhu, 2021). Consequently, models trained on such data risk not only poor generalization but also the inheritance of these data-induced flaws, which can be amplified during adaptation to downstream tasks (Frénay & Verleysen, 2013; Song et al., 2022). A critical goal of fine-tuning is therefore to learn new capabilities while mitigating the effects of this "Catastrophic Inheritance".

A.2 Mitigating Bias through Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) methods offer a promising foundation for mitigating catastrophic inheritance. By design, adapting models with minimal parameter updates can theoretically limit overfitting to inherited noise and help preserve foundational knowledge (Houlsby et al., 2019; Zaken et al., 2021; Lester et al., 2021). However, as we argue in the main paper, this promise is not fully realized in practice. Techniques like low-rank adaptations (LoRA) (Hu et al., 2021) introduce their own inductive biases, such as the low-rank bottleneck, which can inadvertently exacerbate the very issues they are meant to solve by amplifying spurious correlations. This critical gap motivates the development of more principled, explicit regularization techniques—like those proposed in our work—that are tailored to the unique challenges of the PEFT paradigm.

A.3 Typologies of Noise in Pre-training Data

The vast web-scale corpora used to train modern language models, such as LLaMA-2 (Touvron et al., 2023) and GPT-4 (OpenAI, 2023), inevitably contain significant noise and distributional biases. The sheer scale of these datasets makes comprehensive manual curation impractical, meaning models are often exposed to duplicated, corrupted, or irrelevant information during pre-training (Elazar et al., 2023; Birhane & Prabhu, 2021). When fine-tuned, these models can struggle to distinguish signal from noise, which in turn degrades downstream performance. Understanding the specific typologies of this data-induced noise is therefore crucial for developing more robust models. We categorize the primary challenges as follows.

Low-Quality Data This category stems from the uncurated nature of web data. A key issue is **data duplication**, where near-identical content can lead to model overfitting and privacy leakage risks (Carlini et al., 2022; Hernandez et al., 2022). Another challenge is **data corruption**, where inconsistent or erroneous inputs degrade model robustness and performance (Fan et al., 2024; Caswell et al., 2021). Furthermore, **test set contamination**, the leakage of evaluation data into the training corpus, can lead to inflated performance metrics and invalidate a model's evaluation (Roberts et al., 2023; Schaeffer, 2023).

Distributional Skew This form of bias arises from non-uniform data distributions. The most common form is **category imbalance**, where an underrepresentation of certain topics or classes causes the model to perform poorly on those categories, leading to biased or unreliable outputs (Xu et al., 2023; Zhu et al., 2024; Parashar et al., 2024).

Unsafe and Unethical Content Finally, web corpora often contain undesirable content. The presence of **toxic and harmful text**, including offensive, biased, or malicious content, can cause the model to generate inappropriate or harmful outputs, posing significant safety and ethical risks (Zou et al., 2023; Sun et al., 2024).

A.4 Extended Related Work

PEFT and knowledge drift. Beyond classical PEFT methods such as adapters and LoRA (Houlsby et al., 2019; Hu et al., 2021), several recent works explicitly study forgetting and knowledge drift under parameter-efficient adaptation. Smith et al. introduce C-LoRA for continual customization of text-to-image diffusion models and show that naïve LoRA fine-tuning can cause substantial drift across tasks, which they mitigate by carefully constraining adapter updates (Smith et al., 2023). Chen and Garner propose Bayesian parameter-efficient fine-tuning, placing Bayesian priors on LoRA-style adapters to reduce catastrophic forgetting during continual adaptation (Chen & Garner, 2024). These approaches share our goal of stabilizing PEFT, but they mainly operate in parameter space and are

evaluated on diffusion or continual-learning settings. In contrast, BA-LoRA targets catastrophic inheritance in language models and uses three output-space regularizers (consistency, diversity, SVD) that are instantiated in a unified way for both NLU and NLG tasks.

Label noise, covariance, and spectral regularization. Our design is also related to work that analyzes and mitigates label noise in large-scale pre-training. Chen et al. propose a feature-space framework that combines consistency, covariance, and dominant-singular-value regularization to improve robustness to noisy labels in pre-training and demonstrate consistent improvements on downstream tasks (Chen et al., 2023; 2025). Their regularizers act on intermediate representations, whereas BA-LoRA applies analogous ideas directly to the output logits of PEFT-adapted models. More broadly, our diversity regularizer is conceptually aligned with redundancy-reduction objectives such as Barlow Twins (Zbontar et al., 2021) and VICReg (Bardes et al., 2021), which encourage high variance and low cross-covariance to avoid representation collapse. BA-LoRA adapts these principles to the supervised, PEFT setting and combines them with an SVD-based spectral smoothing term, yielding a practical recipe for mitigating catastrophic inheritance in both encoder-only and decoder-only language models.

B Experimental Setup

To rigorously evaluate our proposed method, we conduct a comprehensive set of experiments on a suite of Natural Language Generation (NLG) and Natural Language Understanding (NLU) tasks. Our experimental design, including models, datasets, and training configurations, is detailed below.

B.1 Models

Our evaluation leverages a wide array of pre-trained language models to ensure a comprehensive assessment. For NLG tasks, we primarily utilize large language models renowned for their generative capabilities, including LLaMA-2 (7B, 13B) (Touvron et al., 2023), LLaMA-3 (8B, 70B) (AI@Meta, 2024), Mistral-7B-v0.1 (Jiang et al., 2023), Mixtral-8x7B-v0.1 (Jiang et al., 2024), Gemma-7B (Team et al., 2024), Qwen-1.5-7B (Bai et al., 2023), Yi-1.5-34B (Young et al., 2024), and the Mixture-of-Experts model DeepSeek-MoE-16B (Dai et al., 2024).

For NLU tasks, our experiments employ several key models to investigate different aspects of performance. Our main fine-tuning experiments on the GLUE benchmark utilize DeBERTa-v3-base (He et al., 2021). For the controlled study on pre-training data noise, we specifically select RoBERTa-base (Liu et al., 2019) and T5-base (Raffel et al., 2020) due to their distinct corpus characteristics.

A detailed overview of the primary NLU models is presented in Table 6. These models provide a robust foundation for our study due to their diverse pre-training methodologies. For instance, RoBERTa-base was pre-trained on a high-quality mixed corpus, whereas T5-base was pre-trained on the large-scale and noisier C4 web corpus. DeBERTa-v3-base utilized another diverse dataset with a replaced token detection objective. This architectural and methodological diversity is crucial for a thorough evaluation of our approach.

Table 6: Comparison of pre-trained data and methods for various language models.

Model	Pre-trained Data	Pre-training Method
DeBERTa-v3-base (He et al., 2021)	Wikipedia, BooksCorpus, OpenWebText, CC-News, Stories	Replaced Token Detection with GDES
RoBERTa-base (Liu et al., 2019)	BooksCorpus, English Wikipedia, CC-News, OpenWebText, Stories	Masked Language Modeling
T5-base (Raffel et al., 2020)	Colossal Clean Crawled Corpus (C4)	Text-to-Text Denoising Objective

B.2 Tasks, Datasets, and Metrics

Natural Language Generation (NLG) For NLG, we assess model capabilities across mathematical reasoning, code generation, and instruction following. The benchmarks include GSM8K (Cobbe et al., 2021), MATH (Yu et al., 2023), HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and MT-Bench (Zheng et al., 2024). As summarized in Table 7, evaluation metrics are task-specific: Accuracy for GSM8K and MATH, Pass@1 for HumanEval and MBPP, and GPT-4 based evaluation for MT-Bench.

Table 7: Evaluation metrics for the NLG datasets.

Datasets	GSM8K	MATH	HumanEval	MBPP	MT-Bench
Metric	Accuracy	Accuracy	Pass@1	Pass@1	GPT-4 Evaluation

Natural Language Understanding (NLU) For our NLU evaluation, we utilized the GLUE benchmark (Wang et al., 2018), which comprises a diverse set of tasks. These tasks can be categorized into three groups: two single-sentence classification tasks (CoLA, SST-2), five pairwise text classification tasks (MNLI, RTE, QQP, MRPC, and QNLI), and one text similarity prediction task (STS-B). Following the standard evaluation protocol, we report Matthew's correlation for CoLA, Pearson correlation for STS-B, and accuracy for the remaining tasks. For MNLI specifically, we report both matched and mismatched accuracy.

B.3 Implementation and Training Details

Baseline Comparison For a fair and direct comparison, all baseline results presented in our main experiments are directly obtained from their original publications. Specifically, the NLG baseline results in Table 1 are sourced from the comprehensive study by (Yang et al., 2025). For the NLU benchmarks, the results for DeBERTa-v3-base in Table 2 are taken from (Kang & Yin, 2025). For our BA-LoRA runs, we follow the same core fine-tuning configuration (model, dataset splits, optimizer, learning-rate schedule, batch size, and LoRA rank/placements) as in these works, and only introduce our additional regularization terms with coefficients chosen as described in Section 3.1 and Appendix B.4, to ensure a controlled and equitable evaluation.

For all GLUE experiments, the consistency and diversity regularizers are computed on the fine-tuning data by passing inputs through the pretrained backbone and the BA-LoRA-adapted backbone, using the same shared classification head, to generate teacher and student logits respectively.

MiLoRA vs. BA-LoRA. MiLoRA proposes a spectral variant of LoRA that explicitly exploits *minor* singular components of pretrained weight matrices: instead of parameterizing adapters along the top singular directions (as in PiSSA), MiLoRA allocates capacity to lower-energy directions in weight space, arguing that these under-utilized components can be effective for adaptation while potentially reducing interference with core pretrained knowledge. In contrast, BA-LoRA keeps the underlying LoRA-style parameterization (e.g., standard LoRA, PiSSA, DoRA) and operates entirely in *output space*: it introduces three regularizers on the logits—a consistency term to control knowledge drift, a diversity term to avoid representation collapse, and a spectral SVD term to suppress noisy high-frequency components. Thus, MiLoRA primarily changes *which spectral directions in weight space* are used to represent the adapters, whereas BA-LoRA constrains *how the adapted model behaves* through output-space regularization. These two perspectives are complementary and could, in principle, be combined.

Data Preprocessing for Visualization To analyze the model's feature space under data imbalance, we constructed a custom imbalanced version of the MNLI training dataset. This process began by separating the full training set into three subsets based on their labels. We then retained all samples from the 'entailment' class (100%), while randomly downsampling the 'neutral' class to 10% and the 'contradiction' class to 1% of their original sizes. Finally, these three subsets were concatenated and shuffled to form the training set for the visualization model, thereby simulating a scenario with a highly skewed label distribution.

t-SNE Visualization For the t-SNE visualization, we fine-tuned a RoBERTa-base model for 3 epochs on the imbalanced MNLI training set described above. Subsequently, we extracted the '[CLS]' token representations from the final hidden layer for all samples in the original, balanced MNLI validation set. These high-dimensional features were projected into two dimensions using the t-SNE algorithm with a perplexity of 30, 1000 iterations, and a fixed random seed (42) for reproducibility. The quality of the resulting clusters was also quantitatively assessed using the silhouette score with a cosine distance metric.

Evaluation Frameworks For evaluation, we employed publicly available frameworks. The model's code generation capabilities were assessed using datasets like HumanEval and MBPP through the BigCode Evaluation Harness[2]. Instruction-following performance was evaluated using MT-Bench[3].

B.4 Hyperparameter Settings

NLG (LLaMA-2-7B) Our Natural Language Generation (NLG) experiments involved fine-tuning the LLaMA-2-7B model on a 100,000-sample subset of the MetaMath dataset. The model was trained for a single epoch using BFloat16 (bf16) precision, a maximum sequence length of 512, and an effective batch size of 32, achieved with a per-device batch size of 4 and 4 gradient accumulation steps. For optimization, we employed the AdamW optimizer with a learning rate of 2×10^{-5}, no weight decay, and a cosine learning rate schedule with a 3% warm-up phase. The base LoRA configuration featured a rank (r) of 128, an alpha (α) of 128, and no dropout, with adapters applied comprehensively to the 'q_proj', 'k_proj', 'v_proj', 'o_proj', 'gate_proj', 'up_proj', and 'down_proj' layers. For our proposed BA-LoRA method, we set the regularization coefficients to $\lambda_1 = 0.025$, $\lambda_2 = 0.005$, and $\lambda_3 = 0.005$. The primary coefficient, λ_1, also followed a cosine schedule, while the lambda focus schedule was set to 'two_phase' with a 0.2 warm-up and 0.05 ramp-up ratio. Additional parameters for the SVD-based components included an SVD rank ('svd_k') of 10, an entropy top-k of 20, a distillation temperature of 2.0, and the use of the Frobenius norm for SVD normalization.

NLU (GLUE Benchmark) Our Natural Language Understanding (NLU) experiments on the GLUE benchmark involved three models, each with specific hyperparameter configurations as detailed below.

DeBERTa-v3-base We fine-tuned DeBERTa-v3-base on the GLUE tasks using the AdamW optimizer with a linear learning rate schedule. To strictly align with the PiSSA baseline, we adopted a set of task-specific hyperparameters. The LoRA rank (r) was consistently set to 8 across all tasks. Other key hyperparameters, including the number of epochs, batch size, learning rate, and LoRA alpha, were individually configured for each dataset. The precise configurations are detailed in Table 8. For encoder-only NLU models (e.g., DeBERTa-v3-base on GLUE), we attach a task-specific linear classification head and compute $\mathbf{Z}_P$ and $\mathbf{Z}_F$ by passing the same fine-tuning inputs through the pretrained (teacher) encoder and the BA-LoRA–adapted (student) model, respectively, using the shared classification head, matching the notation used in Sec. 2.2.1.

Table 8: Fine-tuning hyperparameters for the DeBERTa-v3-base model on each task of the GLUE benchmark. The settings are aligned with the PiSSA baseline.

Dataset	Epochs	Batch Size	Learning Rate	LoRA Alpha
MNLI	5	16	5×10^{-4}	8
SST-2	20	16	3×10^{-5}	8
MRPC	20	32	2×10^{-4}	8
CoLA	20	16	1×10^{-4}	8
QNLI	10	32	1×10^{-4}	16
QQP	10	16	1×10^{-4}	8
RTE	50	16	1×10^{-4}	8
STS-B	20	8	3×10^{-4}	8

T5-Base In our experiments with T5-Base, we fine-tuned all models for a single epoch using FP32 precision, a maximum sequence length of 128, and a batch size of 32. Optimization was performed with the AdamW optimizer (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, and no weight decay), coupled with a learning rate of 1×10^{-4}. The learning rate schedule incorporated a 3% warm-up phase followed by a cosine decay. For the LoRA configuration, we set the rank (r) to 8, alpha (α) to 16, and applied it to all linear modules except for the embedding, layer normalization, and language model head layers.

RoBERTa-base For fine-tuning RoBERTa-base on the GLUE benchmark, our setup aligns with standard practices for LoRA-based methods. We employed the AdamW optimizer with a linear

[2] https://github.com/bigcode-project/bigcode-evaluation-harness

[3] https://github.com/lm-sys/FastChat

learning rate schedule, preceded by a warm-up phase over the first 6% of the total training steps. The LoRA configuration was kept consistent across all tasks: the rank (r) was set to 8 for the query (q) and value (v) projection matrices, and the alpha (α) was set to 8. The maximum sequence length was fixed at 512 tokens. Other crucial hyperparameters, including the number of epochs, batch size, and the peak learning rate, were individually tuned for each GLUE task to ensure optimal performance. The precise per-task configurations are detailed in Table 9.

Table 9: Task-specific hyperparameters for fine-tuning RoBERTa-base with LoRA on the GLUE benchmark.

Hyperparameter	**MNLI**	**SST-2**	**MRPC**	**CoLA**	**QNLI**	**QQP**	**RTE**	**STS-B**
Batch Size	16	16	16	32	32	16	32	16
# Epochs	30	60	30	80	25	25	80	40
Learning Rate	5×10^{-4}	5×10^{-4}	4×10^{-4}	4×10^{-4}	4×10^{-4}	5×10^{-4}	4×10^{-4}	4×10^{-4}

C More Experiments

Table 10: Impact of the proposed regularization framework on various LoRA-style methods, evaluated on LLaMA-2-7B. "Reg" denotes the application of our three regularization terms. All results are averaged over 3 runs.

Method	**GSM8K**	**MATH**	**HumanEval**	**MBPP**	**MT-Bench**	**Avg**
LoRA	$42.68_{\pm 0.54}$	$5.92_{\pm 0.15}$	$16.80_{\pm 0.38}$	$21.51_{\pm 0.43}$	$4.60_{\pm 0.14}$	18.30
LoRA + Reg	$51.82_{\pm 0.36}$	$8.69_{\pm 0.39}$	$21.03_{\pm 0.58}$	$33.81_{\pm 0.51}$	$4.73_{\pm 0.24}$	24.02
DoRA	$41.77_{\pm 0.74}$	$6.20_{\pm 0.48}$	$16.86_{\pm 0.54}$	$21.60_{\pm 0.49}$	$4.48_{\pm 0.14}$	18.18
DORA + Reg	$52.71_{\pm 0.42}$	$8.23_{\pm 0.27}$	$21.05_{\pm 0.31}$	$34.78_{\pm 0.28}$	$4.96_{\pm 0.22}$	24.35
PiSSA	$51.48_{\pm 0.34}$	$7.60_{\pm 0.18}$	$19.48_{\pm 0.45}$	$23.84_{\pm 0.46}$	$4.92_{\pm 0.07}$	21.46
BA-LoRA (PiSSA + Reg)	$\mathbf{55.86_{\pm 0.35}}$	$\mathbf{9.47_{\pm 0.52}}$	$\mathbf{23.58_{\pm 0.25}}$	$\mathbf{36.86_{\pm 0.31}}$	$\mathbf{5.11_{\pm 0.05}}$	**25.90**

C.1 Generality of the Regularization Framework

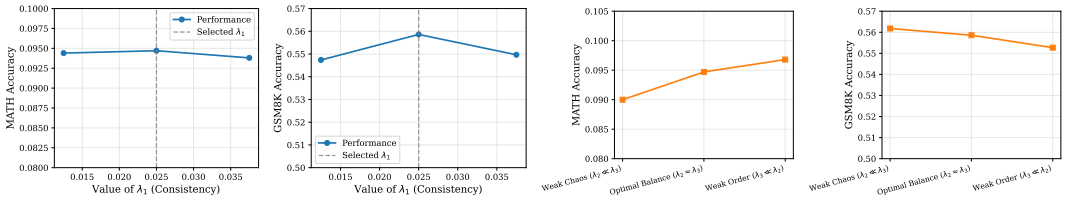
To verify that our regularization framework's benefits extend beyond PiSSA, we integrated it with standard LoRA and DoRA. The results, presented in Table 10, demonstrate the framework's broad applicability and yield a crucial insight. While our regularizers provide substantial performance gains across all tested methods, their effect on standard LoRA is particularly noteworthy. Augmenting standard LoRA with our regularizers is sufficient to match and even surpass the performance of the more advanced PiSSA baseline. This finding underscores that our regularization framework can function as a powerful, model-agnostic enhancement for a wide range of PEFT methods.

Despite the strong standalone performance of the regularizers, the optimal results are consistently achieved by our full BA-LoRA model. This indicates that PiSSA's principled initialization provides a superior foundation upon which our regularization framework can build, leading to the highest overall performance. This validates our integrated approach as the most effective configuration for mitigating catastrophic inheritance and achieving state-of-the-art results.

C.2 Hyperparameter Sensitivity Analysis

To validate the principled selection of our framework's hyperparameters, we conducted a detailed sensitivity analysis. Centered around our final BA-LoRA configuration on LLaMA-2-7B, this study systematically investigates the influence of the core regularization coefficients ($\lambda_1, \lambda_2, \lambda_3$) by perturbing them from their default values. The results, visualized in Figure 4, reveal a broad region of stable performance, supporting the robustness of our chosen configuration.

Sensitivity to the Consistency Anchor (λ_1) As illustrated in Figures 4(a,b), we vary λ_1 across $\{0.0125, 0.025, 0.0375\}$ while keeping $\lambda_2 = \lambda_3 = 0.005$. Performance on both MATH and GSM8K remains highly stable across this range, with only minor fluctuations, indicating a broad region of insensitivity. Our chosen value of $\boldsymbol{\lambda_1 = 0.025}$, highlighted as the optimal point in the figure, is



(a) MATH, λ_1 (b) GSM8K, λ_1 (c) MATH, λ_2/λ_3 (d) GSM8K, λ_2/λ_3

Figure 4: Sensitivity analysis of the core BA-LoRA regularization coefficients. Panels (a,b) show the effect of varying the consistency weight λ_1 on MATH and GSM8K separately, and panels (c,d) show the effect of changing the balance between λ_2 and λ_3.

thus empirically validated as a robust setting that balances preserving pre-trained knowledge with acquiring new task-specific capabilities.

Sensitivity to the Symbiotic Balance (λ_2 and λ_3) Next, we investigate the symbiotic balance between the other two regularizers, which govern a trade-off between performance on the two reasoning benchmarks (MATH and GSM8K). As shown in Figures 4(c,d), we compare our final configuration's balanced setting ($\lambda_2 \approx \lambda_3$) against two asymmetric conditions: a "Weak Chaos" setting ($\lambda_2 \ll \lambda_3$), where the structural regularizer (λ_3) dominates, and a "Weak Order" setting ($\lambda_3 \ll \lambda_2$), where the diversity regularizer (λ_2) is dominant. The results reveal a clear trade-off: disrupting the equilibrium leads to specialized improvements on one benchmark at the cost of the other, while the balanced configuration provides strong performance on both.

C.3 Analysis Across Diverse Model Architectures and Scales

To assess the generalizability and robustness of BA-LoRA, we conducted a comparison against LoRA and PiSSA across ten distinct pre-trained models. This set includes models of varying scales (e.g., LLaMA-2-7B up to LLaMA-3-70B) and architectures, featuring both standard dense models and Mixture-of-Experts (MoE) models such as Mixtral-8x7B. All methods were fine-tuned on a blend of reasoning and code datasets (MetaMathQA-100K and CodeFeedback-100K) and evaluated on GSM8K and HumanEval.

As visualized in Figure 5, BA-LoRA typically achieves the best or near-best performance among LoRA-style methods on both benchmarks, across most model families and scales. The gains over LoRA and PiSSA are especially pronounced on several mid- and large-scale models, suggesting that our regularization framework remains effective beyond the LLaMA-2-7B setting.

Furthermore, this performance advantage largely carries over to computation-constrained settings. The figure also plots the performance of 4-bit quantized versions of each method (QLoRA, QPiSSA, and QBA-LoRA). The overall trend is similar: QBA-LoRA generally matches or exceeds the other quantized baselines, indicating that the benefits of our framework are robust to quantization and remain useful for resource-efficient deployment.

C.4 Performance Analysis Across Different Ranks

We analyzed the performance of BA-LoRA, PiSSA, and LoRA across a range of ranks (from 1 to 128) on the LLaMA-2-7B and Mistral-7B-v0.1 models. Each method was fine-tuned for one epoch on the MetaMathQA-100K dataset and evaluated on GSM8K and MATH. The results, presented in Figure 6, show that BA-LoRA consistently outperforms both LoRA and PiSSA across all ranks, models, and tasks, demonstrating its stable and universal superiority. Furthermore, both BA-LoRA and PiSSA exhibit the remarkable ability to surpass the performance of full fine-tuning at higher ranks, with BA-LoRA often achieving this milestone at relatively low ranks (e.g., rank 16-32). This highlights the strong regularization effect of our approach, as standard LoRA consistently lags behind the full fine-tuning baseline. Moreover, the performance advantage of BA-LoRA over its counterparts is even more pronounced on the Mistral-7B-v0.1 model, suggesting its benefits generalize effectively across different foundational model architectures. These results collectively validate BA-LoRA as a highly efficient and superior fine-tuning method.

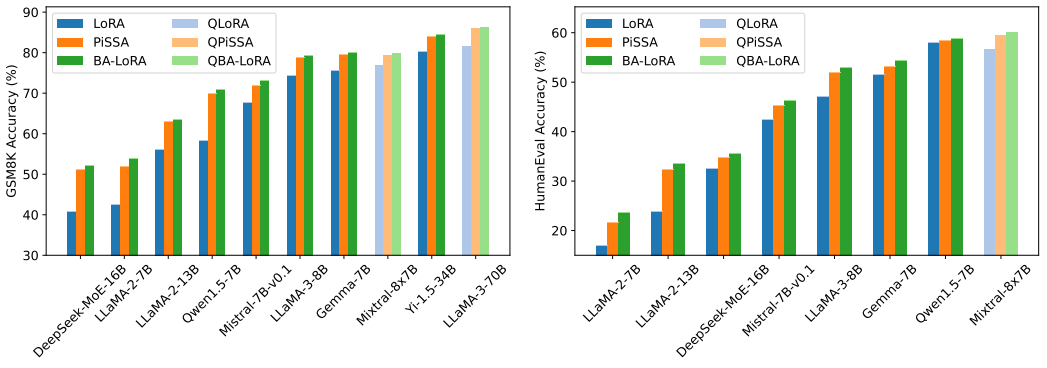


Figure 5: Performance comparison of different models on the GSM8K and HumanEval benchmarks.

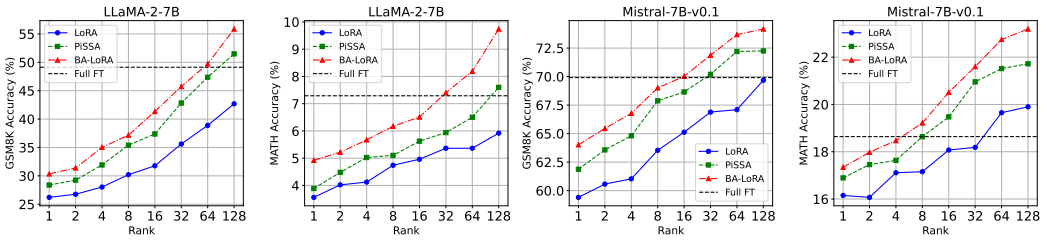


Figure 6: Performance comparison of full fine-tuning, LoRA, PiSSA, and BA-LoRA across different ranks.

C.5 Comparison of Regularization Objectives for NLU

To empirically validate our design choice of Covariance Regularization ($\mathcal{L}_{DR,NLU}$) over Entropy Regularization (commonly used in NLG) for discriminative tasks, we conducted a comparative study on the MNLI benchmark. While entropy minimization is effective for generation, we hypothesized that applying it sample-wise to noisy or imbalanced NLU data may inadvertently encourage the model to be "confidently wrong" on minority classes, rather than effectively preventing representation collapse. The results presented in Table 11 support this hypothesis. Notably, substituting our covariance-based regularizer with an entropy-based one results in performance (90.41%) that is not only inferior to our method (91.26%) but also falls below the standard LoRA baseline (90.71%). This finding confirms that batch-wise feature decorrelation is a structurally superior strategy for mitigating representation collapse in discriminative fine-tuning.

Table 11: Comparison of diversity regularization objectives on MNLI (DeBERTa-v3-base).

Method	MNLI Accuracy
Standard LoRA	90.71
BA-LoRA (w/ Entropy Reg.)	90.41
BA-LoRA (w/ Covariance Reg.)	**91.26**

C.6 Quantitative Analysis of Cluster Quality and Minority Representation

To complement the visual analysis provided in Figure 3 (main text), we conduct a quantitative evaluation of the feature space separation under the imbalanced fine-tuning setting (MNLI task). We compute metrics in the original high-dimensional logit space (before t-SNE projection) to avoid dimensionality reduction artifacts. We report the **Global Silhouette Score** (over all classes), the **Minority-Class Silhouette Score** (measuring the isolation of the minority cluster), and the **Minority-Class Recall** (measuring classification accuracy on the minority class).

Table 12 presents the results. Standard LoRA exhibits a near-zero minority silhouette score (0.015) and a trivial minority recall (5.8), quantitatively confirming the representation collapse observed visually. In contrast, BA-LoRA achieves a significantly higher minority silhouette score (0.425) and restores the minority recall to 61.7. These results demonstrate that BA-LoRA effectively mitigates the catastrophic inheritance of pre-trained priors, enabling the model to establish a distinct and semantically meaningful decision boundary for the minority class.

Table 12: Quantitative analysis of feature separation and classification performance on the minority class (MNLI imbalanced setting). Metrics are computed in the high-dimensional logit space.

Method	**Global Silhouette** (All Classes)	**Minority-Class Silhouette** (Cluster Quality)	**Minority-Class Recall** (Accuracy (%))
Standard LoRA	0.207	0.015	5.8
PiSSA	0.247	0.128	26.4
BA-LoRA (Ours)	**0.351**	**0.425**	**61.7**

D More Discussions

Here, we offer further insights into our work.

D.1 Practical Hyperparameter Guidelines

For a new model–task configuration, BA-LoRA can be tuned with a simple recipe, without fragile fine-grained search. First, choose initial values for $\lambda_{\text{consistency}}$, $\lambda_{\text{diversity}}$, and λ_{svd} such that, on a small calibration batch, each regularization term contributes **a meaningful fraction to the total loss (ensuring gradient magnitudes are commensurate with the task loss)**, so that no component is inactive or overwhelmingly dominant. Second, keep the ratios between the three coefficients fixed and tune a single global scale λ_{global} that multiplies all of them, i.e., $\lambda_i \leftarrow \lambda_{\text{global}}\lambda_i$ for $i \in \{\text{consistency}, \text{diversity}, \text{svd}\}$; in our experience, a coarse search over a few values (e.g., $\{0.5, 1.0, 2.0\}$) on a held-out validation split is sufficient. Third, if needed, individual coefficients can be adjusted by coarse factors (such as $\times 0.5$ or $\times 2$) to emphasize preserving pretrained knowledge (larger $\lambda_{\text{consistency}}$), avoiding collapse (larger $\lambda_{\text{diversity}}$), or suppressing noisy high-frequency components (larger λ_{svd}).

As shown by the ablation study in Section 3.2.4 and the sensitivity analysis in Appendix C.2, BA-LoRA achieves stable performance around these configurations and consistently outperforms vanilla LoRA and PiSSA in this region, suggesting that this lightweight procedure is sufficient in practice. Moreover, for each backbone we fix a single configuration of the auxiliary hyperparameters (s, T, K, k) based on a small validation experiment and reuse it across all datasets, avoiding per-benchmark tuning. **Given the stability observed in our preliminary experiments, and to maintain efficiency**, we use these as reasonable per-backbone defaults rather than performing an exhaustive, computationally expensive sensitivity sweep.

D.2 Additional Discussion on RoBERTa vs. T5

Section 3.2.2 uses RoBERTa-base and T5-base as a natural comparison for probing robustness to inherited noise, since RoBERTa is trained on a curated mixture of corpora while T5 is pretrained primarily on the C4 web corpus. However, these models also differ in architecture (encoder-only vs. encoder–decoder) and pre-training objective (masked LM vs. text-to-text denoising). As a result, the larger BA-LoRA gains we observe on T5 should be interpreted as evidence that is compatible with our noisy-pretraining hypothesis, rather than a fully controlled causal test. A more definitive analysis would require models with closely matched architectures and objectives but systematically varied pre-training corpora, which we leave for future work.

D.3 Discussion on the Choice of Regularization Targets

A key design choice in BA-LoRA is the application of regularization terms in the model's output space (i.e., on logits and their derived distributions) rather than directly on the trainable adapter parameters (A and B). This section provides further justification for this principled decision.

Regularizing the low-rank adapter weights directly, for instance, by penalizing the norm of A or B, is a viable alternative. However, this approach presents a significant challenge: the mapping from the low-dimensional parameter space of the adapters to the high-dimensional functional space of the model's final output is highly complex and non-linear. Consequently, a simple constraint on the adapter weights (e.g., a small norm) does not guarantee the desired functional behavior (e.g., output diversity or consistency with the pre-trained model). The effect of such parameter-space regularization on the final model output is often unpredictable and difficult to control.

In contrast, applying regularization directly in the output space offers a more direct and interpretable path to achieving our goals. By directly penalizing undesirable properties in the output logits or probability distributions—such as their deviation from the pre-trained model (Knowledge Drift), their lack of diversity (Representation Collapse), or their over-reliance on non-robust features (Overfitting to Noise)—we are explicitly constraining the model's final behavior. This approach ensures that our optimization objective is perfectly aligned with the ultimate goal of mitigating the functional consequences of Catastrophic Inheritance. The strong and consistent performance of our framework across diverse models, tasks, and ranks, as demonstrated in our experiments, serves as powerful empirical validation for this output-space regularization strategy.

D.4 Conceptual Foundations and Synergy of Regularizers

The three regularization terms proposed in BA-LoRA—consistency, diversity, and SVD-based regularization—were not chosen arbitrarily. Each is inspired by well-established principles in the machine learning literature for improving model robustness and generalization, and they are designed to work in synergy.

Origins. The **Consistency Regularizer** (implemented as a KLD-based distillation loss in our experiments) is a form of knowledge distillation (Hinton et al., 2015), specifically self-distillation, where the pre-trained model acts as the teacher. The **Diversity Regularizer** is rooted in principles from representation learning and information theory. The covariance-based term for NLU is directly inspired by methods that combat representation collapse in self-supervised learning (Bardes et al., 2021), while the entropy-based term for NLG is a classic technique to prevent mode collapse and improve diversity in generative models (Cover, 1999). Finally, the **SVD Regularizer** builds upon the principle of spectral regularization, where the singular value spectrum of a weight or feature matrix is constrained to improve generalization. The insight that dominant singular values capture the most robust data patterns is a recurring theme in robust machine learning and transfer learning (Chen et al., 2019). Throughout our experiments we follow the standard temperature-scaled distillation convention (Hinton et al., 2015) and multiply the KL-based distillation loss by T^2. Since dividing logits by T scales the gradients of the KL divergence approximately as $1/T^2$, this factor keeps the effective gradient norm of the consistency loss roughly invariant to T, so that changing T primarily controls the softness of the teacher distribution rather than unintentionally reweighting the regularizer.

Synergy. While each regularizer addresses a distinct failure mode, their combination creates a synergistic effect. For instance, solely enforcing consistency ($\mathcal{L}_{\text{CR}}$) might excessively constrain the model, preventing it from fully adapting to the downstream task. However, when combined with the diversity regularizer ($\mathcal{L}_{\text{DR}}$), the model is encouraged to explore new, diverse representations within the bounds of the pre-trained knowledge. Similarly, the SVD regularizer ($\mathcal{L}_{\text{SVDR}}$) helps ensure that the diverse representations learned are also the most robust and generalizable ones, preventing the model from learning spurious correlations encouraged by a simple diversity objective. Our ablation study (Section 3.2.4) empirically confirms this synergy, showing that the performance of the full BA-LoRA model surpasses the sum of the individual components' contributions.

D.5 On Applying Representation Learning Principles during Fine-Tuning

A key consideration for our work is whether incorporating principles from self-supervised learning (SSL), such as our diversity regularizer, during fine-tuning could disrupt the model's pre-trained representations. We contend that our framework effectively mitigates this risk through two primary mechanisms.

First, the PEFT paradigm, specifically LoRA, inherently limits the scope of any changes. With the vast majority of parameters frozen, the model's core representational geometry remains anchored. Our regularizers guide only the small perturbations introduced by the low-rank adapters, ensuring these updates refine rather than overwrite the foundational knowledge.

Second, our regularization scheme is synergistic. The consistency regularizer ($\mathcal{L}_{\text{CR}}$) acts as a crucial counterweight to the diversity regularizer ($\mathcal{L}_{\text{DR}}$). While $\mathcal{L}_{\text{DR}}$ encourages adaptation and prevents representation collapse on the downstream task, $\mathcal{L}_{\text{CR}}$ ensures this adaptation does not stray from the pre-trained model's robust knowledge manifold. It is precisely this calibrated balance—what we term "guided exploration within a trusted neighborhood"—that allows BA-LoRA to enhance task-specific performance without inducing catastrophic forgetting.

D.6 Limitations and Future Work

While this study validates the effectiveness of BA-LoRA, there are areas for future research. Our empirical evaluation has primarily focused on English-language benchmarks, which are a robust foundation for BA-LoRA; however, future work should extend this validation to multilingual settings and specialized domains to ensure broader applicability of the method. Moreover, our analysis of noisy pre-training in Sec. 3.2.2 relies on RoBERTa-base and T5-base checkpoints, which differ in both architecture and **pre-training objective**; results should be interpreted as suggestive rather than controlled evidence about pre-training noise, and a more definitive study with a fixed architecture pre-trained under systematically varied noise levels is an important direction for future work. In addition, while BA-LoRA's regularization components have shown strong promise, task-specific adaptations could further optimize their performance across a wider range of applications, and exploring such adjustments will be valuable for enhancing robustness and adaptability in diverse use cases.

D.7 Ethics Statement

This study aims to develop and evaluate BA-LoRA, a novel parameter-efficient fine-tuning method designed to mitigate bias and enhance the performance of LLMs. By aiming to create more robust and less biased models, a primary ethical motivation of this work is to contribute to safer and more reliable AI systems. Our research utilizes existing open-source public datasets for both fine-tuning and evaluation purposes. For Natural Language Generation tasks, we employed widely recognized datasets within the research community, including MetaMathQA, CodeFeedback, and WizardLM-Evol-Instruct. These datasets have no known ethical concerns. For Natural Language Understanding tasks, we utilized the GLUE benchmark, standard evaluation dataset in machine learning. We are committed to the responsible development and application of AI technologies. Throughout this research, we will continue to monitor and address any ethical issues that may arise.

D.8 Reproducibility

To ensure the reproducibility of our results, we provide a detailed description of our experimental setup in Section 3.1 and Appendix Section B, including model introduction, dataset introduction, hyperparameter configuration, and evaluation procedures. All models and datasets used are publicly available. In addition, we have refined the implementation scripts and fine-tuning strategies to facilitate independent verification. To further facilitate reproducibility, our source code, including scripts to replicate all main experiments, will be made publicly available upon acceptance.

Use of Large Language Models

In the preparation of this manuscript, a large language model (LLM) was utilized as a writing assistant. The LLM's role was strictly limited to improving the clarity, conciseness, and grammatical

correctness of the text. Specifically, it was used for tasks such as rephrasing sentences, suggesting alternative vocabulary, and checking for stylistic consistency. All core scientific ideas, experimental designs, data analyses, and final conclusions were conceived and formulated exclusively by the human authors.