STEERING BACK-PROPAGATION WITH PRIOR INFORMATION IN NATURAL LANGUAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) often struggle when task-relevant prior knowledge is missing or incorrect, leading to overfitting and hallucinations—especially on tasks with ambiguous or sparse data. Simple prompt concatenation can inject priors, but it often yields only marginal gains and may fail to capture the full intent encoded in the priors. We introduce prior-guided tuning, a paradigm that directly embeds natural-language priors into model learning, and propose **P**rior-based **G**radient **E**diting (**PGE**) as a concrete instantiation. PGE computes auxiliary losses for positive (correct) and negative (misleading) prior prompts and adds their difference as an extra term in the gradient update. By shaping gradient updates with this prior-derived signal, PGE steers the model to internalize desired priors and improve task performance. Empirically, PGE outperforms baselines on both a synthetic mathematical expression mapping benchmark and real-world datasets (Jigsaw and BEAD), producing substantial gains in learning efficiency and robustness. Ablations confirm that priors must be presented together with the original training data to be effective, and attention visualizations show that PGE-trained models attend more to prior-relevant tokens. Our code and data will be made publicly available. [1]

1 INTRODUCTION

In recent years, machine learning models, particularly deep learning models, have demonstrated remarkable performance in acquiring knowledge. They acquire knowledge by learning from vast datasets and modeling the underlying probability distributions. However, this learning capability fundamentally relies on the quality and comprehensiveness of the training data. In real-world settings, training data is rarely comprehensive and perfect. Models often face issues such as missing labels, incomplete data, or noisy inputs (Jeong, 2024). Under such challenging conditions, models either struggle to learn patterns from incomplete data or capture spurious correlations, hindering their generalization to new inputs (Gururangan et al., 2020). Large language models (LLMs) are no exception. When finetuned on domain-specific tasks with scarce or misleading training data, they often misinterpret task semantics, generate hallucinations, or lack robustness. In such cases, injecting priors—the professional expertise required to complete specific tasks—into models becomes essential to supplement data and guide models to accomplish tasks. Thus, incorporating accurate priors during model training—especially in scenarios with scarce data or high ambiguity—is critical.

One common approach to injecting priors into models is simple prompt concatenation, which appends manually designed prompts to training examples (Wei et al., 2022; Ouyang et al., 2022; Cui et al., 2024). However, such methods often yield marginal performance gains in challenging scenarios (Chowdhery et al., 2023), and in some cases may even produce counterproductive effects. Specifically, simple prompt concatenation often requires prompts to reappear during inference, which indicates that it does not deeply embed priors into model parameter updates, thereby limiting its capacity to internalize key domain knowledge. Beyond simple prompt concatenation in training, some studies have attempted to introduce prior information during model inference. Nevertheless, these methods do not fundamentally alter model parameters and cannot effectively reuse prior knowledge from previous prompts in subsequent inferences. As a result, they cannot directly integrate rich

[1] https://anonymous.4open.science/r/Prior-based-Gradient-Editing-7236-0802

natural-language priors as guiding constraints in gradient-based optimization, thus failing to steer models toward desired behaviors (Jeong, 2024).

In this work, we introduce prior-guided tuning, a novel paradigm for integrating natural-language priors into LLM fine-tuning without incurring any inference overhead. Prior-guided tuning uses these priors as auxiliary signals during training to enhance performance, eliminating them entirely at inference time. This contrasts with traditional methods, which treat prior-based prompts as part of the learned mapping and require their presence during inference. Building on prior-guided tuning, we propose **P**rior-based **G**radient **E**diting (**PGE**), a technique that directly edits parameter gradients via auxiliary losses derived from natural-language priors. By intervening in backpropagation updates, PGE enables effective learning and internalization of knowledge guided by natural-language priors throughout training—without additional parameters or inference costs. Our main contributions are:

- We propose prior-guided tuning, a new paradigm for embedding natural-language priors into the training process, addressing the prior knowledge deficiency in current LLM adaptation and illustrating why simple prompt concatenation is insufficient.
- We develop **P**rior-based **G**radient **E**diting (**PGE**), a technique that directly guides backpropagation by editing gradients computed from a combination of priors and original inputs, helping models understand prior knowledge and task-specific requirements while reducing inference costs.
- We conduct extensive experiments on synthetic and real-world tasks, along with ablation studies, demonstrating that PGE significantly improves training performance, as well as the attention patterns of models.

2 Related Work

Our approach draws upon and diverges from three main fields—instruction tuning, contrastive learning in natural language processing (NLP), and gradient editing—each providing key ideas and methods that we adapt and extend. Below, we summarize prior advancements in each field and clarify the relationship between our prior-guided tuning and PGE.

2.1 Instruction Tuning

Instruction tuning—the practice of enhancing pretrained models by appending natural-language instructions to training data—has emerged as a powerful paradigm for improving task generalization and aligning with user instructions. Notable early works include T0 (Sanh et al., 2022), FLAN (Wei et al., 2022), InstructBLIP (Dai et al., 2023), and InstructGPT (Ouyang et al., 2022), which assembled large collections of instruction-formatted tasks and showed improvements over many traditional fine-tuning baselines. Subsequent benchmarks like BIG-Bench (Srivastava et al., 2023) and Super-NaturalInstructions (Wang et al., 2022) systematically categorized various instructions, facilitating broader evaluation and training (Jiang et al., 2021). The concept of instruction-based alignment was further advanced by the InstructGPT series (Ouyang et al., 2022; Bai et al., 2022), which combined supervised fine-tuning on human-written instructions with Reinforcement Learning from Human Feedback (RLHF) to make model behavior more aligned with user needs. This paradigm forms the basis of models such as GPT-3 (Brown et al., 2020) and FLAN-T5 (Chung et al., 2024).

Instruction tuning differs from our PGE method in key aspects. While instruction tuning focuses on describing tasks and informing the model of "what" to do, our PGE method emphasizes "how" to utilize prior knowledge. Instruction tuning typically concatenates instructions with training examples, so models learn instructions as inputs along with the examples. This can make it difficult for the model to separately capture instruction guidance and the underlying sample distribution, which may limit the model's ability to effectively leverage instruction guidance. In contrast, our PGE method within the prior-guided tuning framework explicitly incorporates prior knowledge as an auxiliary loss, preserving the original sample loss calculation to guide model parameter updates effectively.

2.2 Contrastive Learning in LLMs

Contrastive learning enhances embedding discrimination by drawing similar examples closer and distancing dissimilar ones, thus improving model robustness and generalization. The seminal work by Chen et al. (2020) established the foundational framework for contrastive learning in computer vision, influencing subsequent applications in NLP. SimCSE (Gao et al., 2021) employed unsupervised dropout views and supervised pairs to boost sentence embeddings. ConSERT (Yan et al., 2021) and DeCLUTR (Giorgi et al., 2021) utilized contrastive augmentations to capture nuanced semantics and rich context. Karpukhin et al. (2020) applied contrastive loss for bi-encoder training in open-domain question answering, while Khosla et al. (2020) leveraged class labels to create tight clusters for enhanced classification. Additional relevant methodologies include CLIP (Radford et al., 2021), which effectively combined image and text representations through contrastive learning.

Contrastive learning and PGE differ in implementation and application. PGE generates auxiliary gradients from natural-language priors to directly influence model parameter updates, while contrastive learning enhances the model's representation by adjusting the embedding space's geometric structure. In terms of applications, contrastive learning excels in representation learning through data augmentation, whereas PGE is particularly well-suited for scenarios with well-defined domain knowledge that can be expressed in text, such as medical diagnosis.

2.3 Gradient Editing

Gradient editing encompasses techniques for manipulating or constraining gradients during or after training to improve multi-task performance or perform targeted model updates. Multi-task learning research has shown that by surgically modifying gradients (e.g., PCGrad (Yu et al., 2020), GradNorm (Chen et al., 2018), and related methods (Sener & Koltun, 2018; Liu et al., 2021)), conflicting objectives can be coordinated, improving optimization and mitigating negative transfer. Additionally, post-hoc model editing techniques like ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), and other frameworks (Sanh et al., 2022) manipulate learned weights to change model behaviors.

Both traditional gradient editing methods and PGE intervene directly in the gradient update process to adjust model behavior, rather than modifying the architecture or the data distribution. However, traditional methods are primarily used in multi-task learning, while PGE focuses on incorporating natural-language priors during training. PGE explicitly uses natural-language priors to define auxiliary losses, which produce additional gradient signals that encourage the model to incorporate the prior knowledge. In contrast, traditional methods like PCGrad coordinate multi-task optimization based on the geometric relationships of task gradients, with their priors being implicit in the gradient distribution.

3 Problem Setup: Prior-Guided Tuning

In this work, we propose prior-guided tuning, a simple yet powerful paradigm for endowing large language models with explicit prior knowledge through natural-language prompts during training.

3.1 Differences from Traditional Learning Methods

Domain-specific expertise or task-required information can be transmitted to models through two channels: the distribution of training data and human-summarized priors. As illustrated in Figure 1, traditional data-driven approaches rely entirely on knowledge encoded in massive data distributions. On the one hand, data inherently suffers from various flaws as discussed earlier; on the other hand, complex knowledge often demands extensive data, incurring high costs in data annotation and model training. In contrast, prior-guided tuning not only leverages the original data distribution but also directly conveys priors to the model via natural language. This explicit signal enables the model to acquire knowledge more effectively without requiring manual synthesis of training data.

The training and inference paths of prior-guided tuning can be summarized as follows. Assume that we have a large language model M with parameters θ and a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the input instance and y_i is the desired output. Under instruction fine-tuning, the training can be represented via $M_\theta(I_i, x_i) = y_i$, where I_i denotes the task instruction. Prior-guided tuning addi-

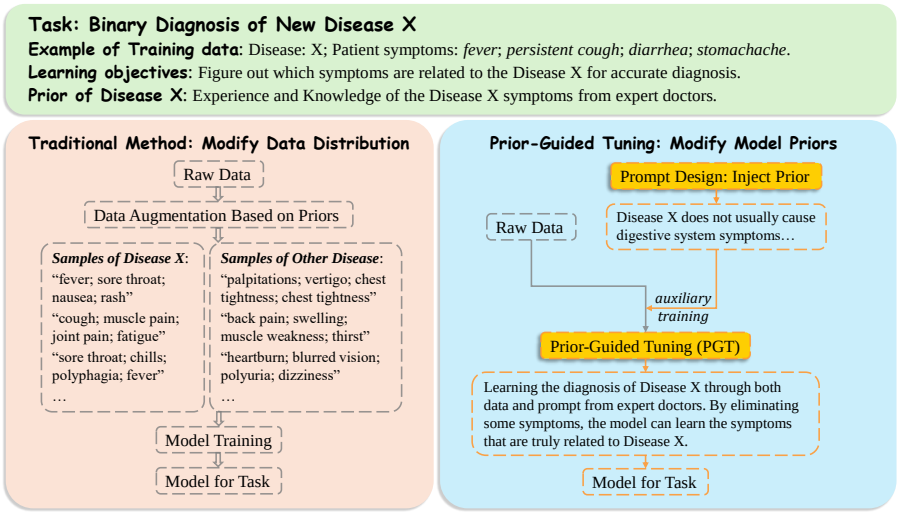

Figure 1: Comparison of conventional data-driven training vs. our prior-guided tuning paradigm in the binary diagnosis of a novel Disease X. prior-guided tuning directly injects expert priors via a natural-language prompt (e.g., "Disease X does not usually cause digestive system symptoms…") and uses it as an auxiliary signal alongside raw data. Specifically, this explicit prior steers the model more efficiently toward the correct hypothesis, without requiring the synthesis of biased datasets.

tionally introduces p_i, which represents the prior knowledge aiding model learning. The inclusion of p_i can supplement knowledge missing from x_i or facilitate the model's learning from x_i. Notably, prior-guided tuning emphasizes that priors serve only as guidance, with the learning still expressed as $M_\theta(I_i, x_i) = y_i$. Thus, the model does not require priors during inference, ensuring it cannot "cheat" by relying on spurious patterns in priors. Instead, the model must internalize these priors into its parameters, which distinguishes PGE from other methods that necessitate priors at inference.

3.2 Synthetic Benchmarks for Evaluation

In real-world scenarios, knowledge embedded in the data distributions and knowledge provided by priors both aid model learning. To quantitatively study the impact of natural-language priors on learning efficiency while excluding the influence of original data distributions, a highly controlled evaluation environment is needed. We introduce a synthetic benchmark based on simple function expression calculation tasks of the form "func(a_1, a_2, …, a_n) = c". In each example, only one parameter determines the answer, while all others are irrelevant. During training, each example is accompanied by a natural-language prior explicitly indicating which parameter to focus on, helping the model identify the critical parameter and map it to the final answer. Crucially, during testing, all prior prompts are removed, and only the original function expressions are presented. Thus, task completion requires the model to internalize the priors into its parameters rather than relying on superficial cues. We instantiate this benchmark in two complementary tasks:

Task 1 The model must learn to select the correct parameter position. Training examples take the form: "[The output of func is its second input parameter.] func(v, v, v, v, v) = v". To eliminate data distribution effects, all five input parameters in mathematical expressions during training have the same value, and the prior explicitly specifies the decisive parameter. This forces the model to understand and follow the prior rather than guessing outputs via co-occurrence frequencies in the training data distribution. During inference, priors are omitted, and five random parameters (e.g., func(v_1, v_2, v_3, v_4, v_5) = v_2) are used to test if the model identifies the correct parameter. The answer is set to the correct parameter itself to simplify the task, and the only challenge is to determine which parameter position is decisive.

Task 2 extends Task 1 by combining parameter selection with arithmetic transformations and language recognition. Each example presents two parameters, one written in Chinese characters and the other in English words, and the mapping is either "add 2 to the Chinese parameter" or "subtract

2 from the English parameter." (e.g., func(22, 26) = 24, where 22 is written in Chinese characters, 26 is written in English words, and the result is given as the Arabic numeral 24). The prior only indicates which parameter to select, while the model learns the add/subtract mapping. During inference, priors are removed, and random parameters (e.g., func(17, 5) = ?) are used; the model must infer which parameter is correct and apply the correct arithmetic rule to it (e.g., 17+2=19 or 5-2=3).

We conducted baseline experiments using plain finetuning (learning entirely from the data distributions) and prompt finetuning (simple prompt concatenation during training). While directly appending prior-based prompts to examples (prompt finetuning) occasionally improved performance, these gains were highly unstable and in some cases this method even degraded performance. This instability motivates our proposed PGE approach, detailed in the next section.

4 Method: Gradient Editing Based on Prompts

4.1 Motivation and Overview

When large language models (LLMs) learn knowledge and perform specific tasks, ensuring that the model follows prior-based natural-language prompts requires it to deeply understand the knowledge and guiding information encoded in the prior. However, Transformer-based models are typically trained to predict the next token using token-level likelihood and updated via backpropagation. This causes the model to potentially learn the token distribution patterns of natural-language priors rather than the deep knowledge encoded in the priors. Under such conditions, simply concatenating prompts with training examples cannot redirect the model's parameter learning towards the knowledge and guidance in the priors.

To address these limitations, we propose **P**rior-based **G**radient **E**diting (**PGE**), a gradient editing strategy that uses natural-language priors to construct auxiliary losses for model learning. Our goal is to integrate prior knowledge into model parameters during training, eliminating the need to reuse these priors at inference time. Since backpropagation is a crucial process for LLM training, influencing backpropagation directly via gradient editing is a natural approach. PGE employs two contrastive prompt forms—positive and negative—based on natural-language priors to assist the model in acquiring domain knowledge and task guidance. Meanwhile, PGE preserves the learning signal from the original samples to avoid large shifts in the model's objective.

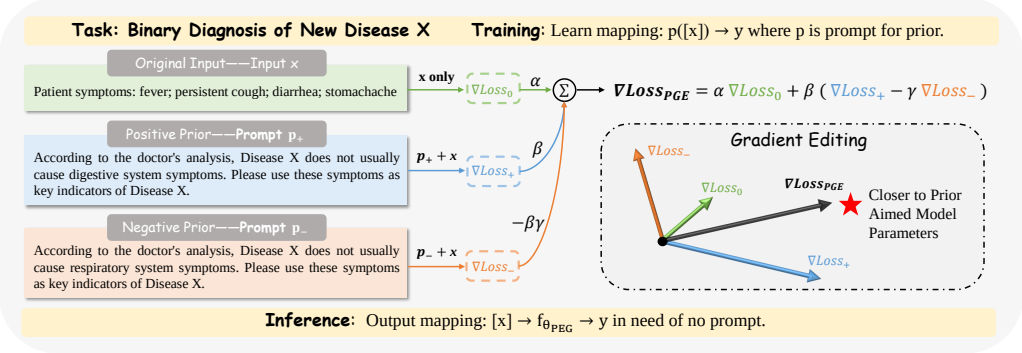

Figure 2: Illustration of Prior-based Gradient Editing (PGE) on a binary diagnosis task for hypothetical Disease X. During training, symptom inputs x are combined with a positive prompt p_+ (true indicators) and a negative prompt p_- (misleading features), yielding three gradient components. These components are aggregated into the training update rule, sculpting parameter changes to ensure correct behavior without inference-time prompts.

4.2 Implementation Details

The standard parameter update gradient for LLM training is expressed as:

$$\nabla_\theta \, \ell(f_\theta([I_i; x_i]), y_i) \tag{1}$$

where ℓ denotes the loss function, θ represents the model parameters, x_i is a training input, and I_i and y_i are the corresponding instruction and output for x_i. As illustrated in Figure 2, PGE aims to add an auxiliary gradient, which is defined as the difference between a positive gradient and a negative gradient:

$$\nabla_\theta \, \ell(f_\theta([p_+; I_i; x_i]), y_i) \; - \; \gamma \, \nabla_\theta \, \ell(f_\theta([p_-; I_i; x_i]), y_i) \tag{2}$$

Here, p_+ contains the correct prior (the desired "positive" prompt), p_- contains the incorrect prior (the undesired "negative" prompt), and the scalar $\gamma > 0$ controls the penalty strength for the negative prompt.

Following Equation (2), we integrate all gradient contributions into a core update rule $\nabla_\theta \mathcal{L}_{\text{PGE}}$:

$$\nabla_\theta \mathcal{L}_{\text{PGE}} \; = \; \alpha \, \nabla_\theta \, \ell(f_\theta([I_i; x_i]), y_i) + \beta \left(\nabla_\theta \, \ell(f_\theta([p_+; I_i; x_i]), y_i) - \gamma \, \nabla_\theta \, \ell(f_\theta([p_-; I_i; x_i]), y_i) \right) \tag{3}$$

where $\alpha, \beta > 0$ are fixed hyperparameters that balance the two objectives. This equation clarifies how the overall gradient decomposes into the combined force of data fitting and prior fitting. Because differentiation is linear, adding loss terms corresponds to adding their gradients. In practice, we simplify the loss as:

$$\mathcal{L}_{\text{PGE}} \; = \; \alpha \, \mathcal{L}_0 + \beta \left(\mathcal{L}_+ - \gamma \, \mathcal{L}_- \right) \tag{4}$$

where $\mathcal{L}_0$, $\mathcal{L}_+$, and $\mathcal{L}_-$ correspond to terms in Equation (3). The standard backpropagation is applied to $\mathcal{L}_{\text{PGE}}$. Notably, the unbounded growth of $\mathcal{L}_-$ during training may cause instability, necessitating optimization strategies such as gradient clipping and upper-bounding the negative loss.

In our synthetic tasks, priors only require the model to focus on specific parameters, so positive and negative priors are manually written and concise. For real-world tasks with more complex training data, both priors are generated by LLMs (DeepSeek-v3 (DeepSeek-AI et al., 2024) and GPT-4o (Hurst et al., 2024)), with similar lengths. Since each dataset involves a single task type, the same set of positive and negative priors is used to avoid label leakage.

5 Results

We finetuned LLaMA 3.1 (8B and 70B) (Dubey et al., 2024; Patterson et al., 2022) and Qwen 2.5 (7B) (Yang et al., 2025) models (Team, 2024; Yang et al., 2024)), using LoRA (Hu et al., 2022) adapters of rank 16 on NVIDIA RTX 4090 and A100 GPUs, updating all weight matrices (q_proj, k_proj, v_proj, o_proj, gate_proj, down_proj and up_proj) except the embedding and output layers/heads. Each model underwent ten epochs of training with learning rates in [1e-4, 5e-4], and the best checkpoint was chosen according to validation performance. All checkpoints used AWQ 4-bit quantization (Lin et al., 2024), and LoRA adapters were concatenated onto the quantized linear projections. We mainly compared three strategies: plain finetuning, which directly updates model parameters on the task data; prompt finetuning, which prepends prior-based prompts to each example; and our PGE method, which integrates positive and negative priors according to Equation 4 and tuned hyperparameters α and β (with γ fixed at 0.1). Notably, we did not use any prior during inference on either the synthetic or the real-world datasets. In addition, the model template, prior prompts for experiments, and the discussion of computational costs are in the Appendix.

The Synthetic Dataset. Table 1 reports the exact match accuracy of Task 1 on the synthetic dataset across five answer positions. Under plain finetuning, most models were biased toward a random option due to the lack of priors, leading to poor performance on other positions. Prompt finetuning yielded negligible or even negative gains compared to plain finetuning. In contrast, PGE achieved significantly higher accuracy on most answer positions. From another perspective, PGE effectively reversed the model's original incorrect prior (i.e., bias towards the first option) by injecting correct priors, promoting balanced performance across all positions. Table 2 summarizes the performance of Task 2 on the synthetic dataset. Similar to Task 1, plain finetuning and prompt finetuning performed poorly, while PGE outperformed both baselines significantly and partially achieved balanced performance in cross-lingual scenarios.

Real-world Datasets. To evaluate the generalizability of our method, we selected the Jigsaw dataset (Do, 2019), which contains real user comments annotated with toxicity and multiple identity terms (including gender). In practical applications, models sometimes over-rely on the association between specific genders and text toxicity (e.g., deeming text toxic upon encountering particular gender terms) or ignore toxicity words that are specific to genders. Thus, explicit priors are crucial

	Hyperparameters		QA performance at 5 positions					
	α	β	**1st**	**2nd**	**3rd**	**4th**	**5th**	**Avg**
LLaMA 3 8B								
Baselines								
Plain finetuning	-	-	**98.0**	31.0	22.0	15.0	**47.0**	42.6
Prompt finetuning	-	-	88.0	24.0	11.0	10.0	29.0	32.4
Ours								
PGE method	0.5	0.5	**98.0**	**51.0**	**84.0**	**32.0**	40.0	**61.0**
LLaMA 3 70B								
Baselines								
Plain finetuning	-	-	**97.0**	24.0	19.0	22.0	26.0	39.8
Prompt finetuning	-	-	78.0	68.0	43.0	**59.0**	37.0	57.0
Ours								
PGE method	0.7	0.3	86.0	**98.0**	**62.0**	47.0	**96.0**	**77.8**
Qwen 2.5 7B								
Baselines								
Plain finetuning	-	-	40.0	36.0	33.0	16.0	33.0	31.6
Prompt finetuning	-	-	38.0	46.0	49.0	**30.0**	27.0	38.0
Ours								
PGE method	0.5	0.5	**53.0**	**59.0**	**56.0**	26.0	**58.0**	**50.4**

Table 1: Exact-match accuracy (%) on the five-argument mapping synthetic benchmark (Task 1) for LLaMA 3.1 (8B and 70B) and Qwen 2.5 7B under plain finetuning, prompt finetuning, and PGE.

	Hyper-parameters		QA performance			
			Position		Language	
	α	β	**1st**	**2nd**	**Ch**	**En**
LLaMA 3 8B						
Baselines						
Plain finetuning	-	-	66.8	40.0	66.8	40.0
Prompt finetuning	-	-	63.9	38.4	47.4	27.9
Ours						
PGE method	0.7/0.3	0.3	**85.8**	**58.3**	**90.0**	**68.4**

	Hyper-parameters		QA performance			
			Position		Language	
	α	β	**1st**	**2nd**	**Ch**	**En**
LLaMA 3 70B						
Baselines						
Plain finetuning	-	-	90.0	50.0	**90.0**	50.0
Prompt finetuning	-	-	88.4	54.7	61.6	60.5
Ours						
PGE method	0.5	0.5	**100.0**	**72.6**	**90.0**	**76.3**

Table 2: Exact-match accuracy (%) on the two-argument bilingual (Chinese/English) mapping synthetic benchmark (Task 2) for LLaMA 3.1 (8B and 70B) under plain finetuning, prompt finetuning, and PGE.

for addressing this issue. To focus on gender bias, we excluded samples that were more strongly associated with labels other than gender (e.g., religion and race) and only retained samples strongly associated with gender in the dataset (defined as at least one gender label score exceeded 0.5). To simulate a prior-free scenario, we used only 30% of these gender-associated samples and randomly split them into 80% training and 20% test. Table 5 shows the accuracy, positive-class F1, negative-class F1, and macro F1 scores by gender. Plain finetuning achieved high overall accuracy but low positive class F1 scores. Prompt finetuning did not improve the performance much and sometimes further worsened it. In contrast, PGE (α=0.7, β=0.3) significantly improved the positive F1 scores for all genders while maintaining or increasing overall accuracy and macro F1 score.

To verify PGE's capability in more general domains, we chose the shainar/BEAD benchmark (Raza et al., 2024), which includes three sub-tasks: bias, sentiment, and toxicity. To ensure comparable data volumes, we down-sampled larger subsets (sentiment and toxicity) to 30,000–40,000 samples to match the bias task and unified the formatting of all datasets. This sampling strategy balanced cross-task data volumes while preserving training conditions devoid of priors. As shown in Table 4, prompt finetuning only provided modest improvements over plain finetuning (even a decline in the sentiment task), whereas PGE consistently outperformed both baselines across all three sub-tasks, demonstrating its generality in incorporating appropriate priors into language models to address diverse real-world classification tasks.

LLaMA 3 8B						*Qwen 2.5 7B*					
Method	Gender	Acc	F1+	F1-	Macro F1	**Method**	Gender	Acc	F1+	F1-	Macro F1
Plain	female	88.9	0.394	0.939	0.667	Plain	female	88.6	0.463	0.937	0.700
	male	87.9	0.391	0.933	0.662		male	88.0	0.492	0.932	0.712
	other	**85.3**	0.546	**0.912**	0.729		other	85.3	0.545	0.912	0.729
	trans	83.7	0.370	0.907	0.639		trans	83.3	0.407	0.903	0.655
Prompt	female	87.4	0.185	0.932	0.558	Prompt	female	88.6	0.478	0.936	0.707
	male	86.3	0.202	0.999	0.564		male	88.2	0.514	0.933	0.724
	other	79.4	0.222	0.881	0.552		other	85.3	0.545	0.912	0.729
	trans	81.8	0.136	0.897	0.568		trans	82.3	0.393	0.896	0.645
Ours	female	**90.6**	**0.587**	**0.947**	**0.767**	Ours	female	**92.1**	**0.647**	**0.956**	**0.801**
	male	**89.5**	**0.575**	**0.940**	**0.758**		male	**91.0**	**0.630**	**0.949**	**0.789**
	other	**85.3**	**0.615**	0.909	**0.762**		other	**91.2**	**0.769**	**0.945**	**0.857**
	trans	**84.7**	**0.500**	**0.910**	**0.705**		trans	**86.6**	**0.533**	**0.922**	**0.728**

Table 3: Test accuracy (%) and F1 score(s) by gender category on the Jigsaw toxicity subset, contrasting plain finetuning, prompt finetuning, and our PGE for LLaMA 3 8B and Qwen 2.5 7B models.

	Hyperparameters		Classification accuracy		
	α	β	**Bias**	**Sentiment**	**Toxic**
LLaMA 3 8B					
Baselines					
Plain finetuning	-	-	80.2	76.8	80.6
Prompt finetuning	-	-	80.2	69.7	80.9
Ours					
PGE method	0.7	0.3	**82.1**	**79.6**	**82.7**

Table 4: Classification accuracy (%) on the BEAD benchmark subtasks (bias, sentiment, toxicity) for LLaMA 3.1 8B under plain finetuning, prompt finetuning, and PGE.

6 Discussion

Supplementary Baselines and Ablation Studies. To further demonstrate the capability of the PGE method, we conducted additional experiments using the LLaMA-3.1-Instruct model. In "Data Augmentation", 10,000 synthetic samples were generated with reference to real data via GPT-4o mini to align with task requirements, and were incorporated into the original training dataset. As shown in Figure 3, synthetic data from large models failed to help the model learn prior knowledge—likely because the synthetic samples depended on original samples—leading to a slight performance degradation. For "Priors without data", we removed the original data that had been combined with the positive or negative priors ($p_+ + x$ and $p_- + x$ in Figure 2 were replaced by p_+ and p_-) and observed a performance decline, indicating that priors must be combined with original training data to be effective.

Attention Visualization. To better understand the reasons behind PGE's performance enhancement, we examined the self-attention behavior of the model on a representative toxic comment from the BEAD dataset. Self-attention patterns essentially reflect how Transformers allocate focus among tokens, and prior studies have linked superior attention distributions to stronger performance (Weston & Sukhbaatar, 2023; Tang et al., 2022). Thus, we compared the token-level attention maps of four model variants: (1) the untrained base model, (2) standard finetuned model, (3) the prompt-tuned model, and (4) our PGE-trained model. To ensure a fair comparison, we first addressed the "attention sink" phenomenon (Xiao et al., 2024), where the start-of-sequence token may dominate normalized attention weights. We removed the contribution of the start-of-sequence token, re-normalized the attention scores of all remaining tokens, and visualized these scores across different sample types.

As shown in Figure 4, unlike all other variants, the PGE-trained model did not consistently allocate excessive attention to initial, semantically empty tokens, thereby preserving its ability to capture core toxic content. Additionally, it also allocated substantial, well-balanced attention to contrastive tokens such as "except" and "exception," which signal a shift in scope and help the model focus

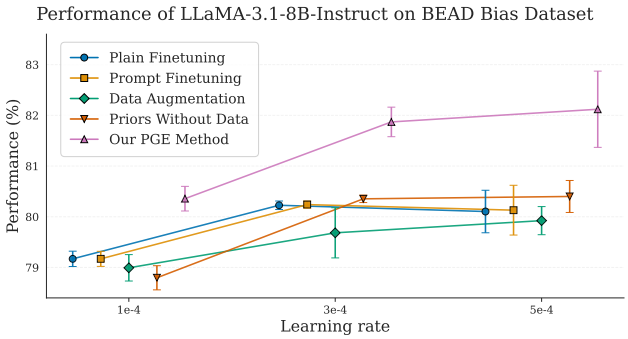

Figure 3: Classification accuracy (%) on the BEAD bias benchmark for five LLaMA 3.1 8B Instruct varients under three learning rates.

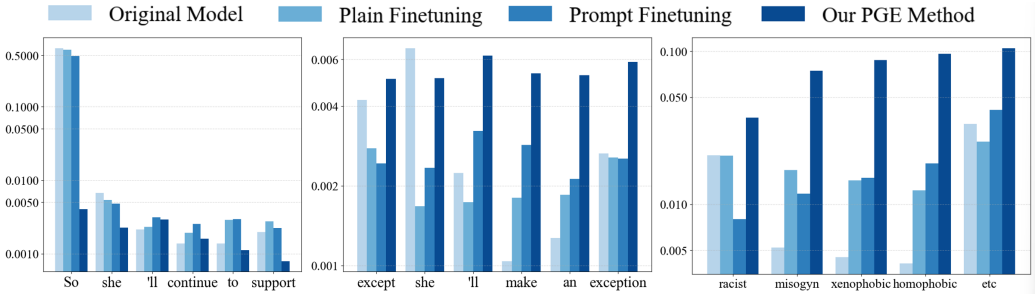

Figure 4: Token-level self-attention distributions for four LLaMA-3.1-Instruct variants on a comment from the BEAD toxic dataset : *So she'll continue to support communities that are different from her own... just as long as those communities don't include people she doesn't agree with politically... and she won't stand for bigotry... except she'll make an exception for the half of the country she believes to be racist, misogynist, xenophobic, homophobic, etc. Got it.*

on the subsequent toxic content. Most crucially, the PGE model consistently allocated higher attention to consecutive toxic tokens (e.g., racist, misogynist, xenophobic, homophobic, etc.), with its attention weight on "etc." reaching 0.105—significantly exceeding the corresponding weights of the other three variants (0.034, 0.026, and 0.041 for original, plain-finetuning and prompt-finetuning), demonstrating its effective focus on toxic information. Collectively, these observations empirically explain why PGE more effectively injects relevant prior knowledge into language models, driving their superior performance in toxic classification tasks.

7 Conclusion

In this paper, we introduce Prior-based Gradient Editing (PGE) under prior-guided tuning paradigm as a principled approach to infusing natural-language priors into large language model training without incurring any inference-time computation cost. PGE shapes the backpropagated gradients by constructing auxiliary losses through positive and negative priors, thereby enhancing the model's performance in learning knowledge and completing tasks. Our experiments on synthetic benchmarks and real-world classification tasks, including the Jigsaw and BEAD datasets, demonstrate that when all priors are removed during testing, PGE enables the model to follow the guidance of explicit priors and consistently outperform plain finetuning and prompt finetuning baselines, achieving significant improvements in macro F1-score and accuracy. Additionally, ablation experiments validate the necessity of combining priors with original data and the limitations of traditional data augmentation methods. Attention visualization analysis further explores the advantages of the PGE method.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022. doi: 10.48550/ARXIV.2204.05862. URL https://doi.org/10.48550/arXiv.2204.05862.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 793–802. PMLR, 2018. URL http://proceedings.mlr.press/v80/chen18a.html.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. URL https://jmlr.org/papers/v24/22-1144.html.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024. URL https://jmlr.org/papers/v25/23-0870.html.

Jiaxi Cui, Wentao Zhang, Jing Tang, Xudong Tong, Zhenwei Zhang, Amie, Jing Wen, Rongsheng Wang, and Pengfei Wu. Anytasktune: Advanced domain-specific solutions through task-fine-tuning. *CoRR*, abs/2407.07094, 2024. doi: 10.48550/ARXIV.2407.07094. URL https://doi.org/10.48550/arXiv.2407.07094.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9a6a435e75419a836fe47ab6793623e6-Abstract-Conference.html.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024. doi: 10.48550/ARXIV.2412.19437. URL https://doi.org/10.48550/arXiv.2412.19437.

Quan Do. Jigsaw unintended bias in toxicity classification. 2019.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 6894–6910. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.552. URL https://doi.org/10.18653/v1/2021.emnlp-main.552.

John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader. Declutr: Deep contrastive learning for unsupervised textual representations. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 879–895. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.72. URL https://doi.org/10.18653/v1/2021.acl-long.72.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.740. URL https://doi.org/10.18653/v1/2020.acl-main.740.

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. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024. doi: 10.48550/ARXIV.2410.21276. URL https://doi.org/10.48550/arXiv.2410.21276.

Cheonsu Jeong. Fine-tuning and utilization methods of domain-specific llms. *CoRR*, abs/2401.02981, 2024. doi: 10.48550/ARXIV.2401.02981. URL https://doi.org/10.48550/arXiv.2401.02981.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know *When* language models know? on the calibration of language models for question answering. *Trans. Assoc. Comput. Linguistics*, 9:962–977, 2021. doi: 10.1162/TACL_A_00407. URL https://doi.org/10.1162/tacl_a_00407.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12,*

2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Guangxuan Xiao, and Song Han. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. *GetMobile Mob. Comput. Commun.*, 28(4):12–17, 2024. doi: 10.1145/3714983.3714987. URL https://doi.org/10.1145/3714983.3714987.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 18878–18890, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

David A. Patterson, Joseph Gonzalez, Urs Hölzle, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28, 2022. doi: 10.1109/MC.2022.3148714. URL https://doi.org/10.1109/MC.2022.3148714.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Shaina Raza, Mizanur Rahman, and Michael R Zhang. Beads: Bias evaluation across domains. *arXiv preprint arXiv:2406.04220*, 2024.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference*

on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 525–536, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e345e339f35a30c8f65edce-Abstract.html.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartlomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cèsar Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan J. Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, François Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse H. Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, José Hernández-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory W. Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, María José Ramírez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael I. Ivanitskiy, Michael Starritt, Michael Strube, Michal Swedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T.,

Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Milkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima (Shammie) Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay V. Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=uyTL5Bvosj.

Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=fR-EnKWL_Zb.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 5085–5109. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.340. URL https://doi.org/10.18653/v1/2022.emnlp-main.340.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Weston and Sainbayar Sukhbaatar. System 2 attention (is something you might need too). *CoRR*, abs/2311.11829, 2023. doi: 10.48550/ARXIV.2311.11829. URL https://doi.org/10.48550/arXiv.2311.11829.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 5065–5075. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.393. URL https://doi.org/10.18653/v1/2021.acl-long.393.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report. *CoRR*, abs/2501.15383, 2025. doi: 10.48550/ARXIV.2501.15383. URL https://doi.org/10.48550/arXiv.2501.15383.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

A Appendix

A.1 Data Details

We constructed our synthetic benchmark to rigorously assess a model's ability to internalize natural-language priors in two scenarios. In Task 1, each example comprises five numerical parameters with an identity mapping—i.e., "func(a_1, a_2, …, a_n) = c"—and an explicit prior instructing the model which position to select. Task 2 extends this setup to two bilingual parameters (one tagged in Chinese, the other in English) combined with a simple arithmetic operation (either "add 2" or "subtract 2").

An example of Task 1 under LLaMA 3.1 8B & 70B Instruct template:

$< |start\_header\_id| > system < |end\_header\_id| >$
Cutting Knowledge Date: December 2023 Today Date: 26 Jul 2024
Provide the output only without steps.$< |eot\_id| >$
$< |start\_header\_id| > user < |end\_header\_id| >$
func(8,8,8,8,8)=$< |eot\_id| >< |start\_header\_id| >$

$assistant < |end\_header\_id| >$
$8 < |eot\_id| >$

By design, these tasks isolate the challenges of parameter selection and arithmetic transformation under instruction guidance. We find that LLaMA 3.1 exhibits a distinct preference for the first position, for longer parameter names, and for addition mappings with small values; Qwen 2.5, which is trained on substantially more Chinese data, shows a stronger bias toward Chinese-tagged parameters but otherwise shares similar positional and mapping preferences. In Task 2, we therefore paired LLaMA 3.1's favored English label with the second argument to counterbalance its first-position bias, while Qwen 2.5's overlapping preference for Chinese and the first slot made it particularly challenging for that model to attend to the second parameter.

All training and test examples were generated via a shared template mechanism (the code for which is publicly available in our repository), ensuring that aside from the removal of prompts at inference time, the format remains identical across phases. During evaluation, we present the model with raw function calls in a zero-shot setting—no priors are supplied—so that correct outputs attest to fully internalized instruction knowledge.

The positive (correct) and negative (misleading) prior prompts for Task 1 in the synthetic dataset (assuming that the first parameter is the correct parameter) are:

> *Func is a function with five input parameters. The output of func is determined by its first input parameter.*

> *Func is a function with five input parameters. The output of func is determined by its second / third / fourth / fifth input parameter.*

Each time, one negative (misleading) prior prompt is randomly selected from the four options for PGE training. Except that in Task 2 the function "func" contains only two parameters — so there are no "third", "fourth", or "fifth" descriptions — the prior prompt settings for the rest remain the same as in Task 1 in the synthetic dataset.

The positive (correct) and negative (misleading) prior prompts for the Jigsaw dataset are:

> *You are a toxicity classification assistant. Please judge whether the following comment text is toxic based on its context and actual meaning, not merely on identity-related terms—especially gender. Prioritize catching all toxic content and minimizing any missed toxic cases. If it is toxic, output 1; if it is not toxic, output 0.*

> *You are a toxicity classification assistant. Please judge whether the following comment text is toxic solely by checking for the presence of identity-related or sensitive terms (especially gender words). If you find any gender-related term, immediately output 1; otherwise, output 0.*

The positive (correct) and negative (misleading) prior prompts for the BEAD bias dataset are:

> *Detect bias by checking for: (a) emotionally loaded words, (b) sweeping generalizations, (c) attacks on people/groups, (d) claims without evidence. Output 1 if detected, else 0.*

> *Label the text as unbiased (0) if it contains emotionally intense words, sweeping claims about groups, or assertions without data. Label as biased (1) if the text is calm and logical.*

The positive (correct) and negative (misleading) prior prompts for the BEAD sentiment dataset are:

> *Analyze the text's sentiment (Positive/Negative) by detecting explicit emotions, contextual tone, and author's intent. Focus on strong indicators: love/hate expressions, sarcasm marked by quotes or contradictions, and overall stance toward subjects. Prioritize deeper communicative purpose over isolated words.*

> *Ignore context and sarcasm when classifying sentiment. Rely solely on isolated words while inverting emotional valence: interpret praise as criticism and complaints as approval, treating metaphors literally without considering contextual meaning.*

The positive (correct) and negative (misleading) prior prompts for the BEAD toxic dataset are:

> *Classify the text as TOXIC if it contains insults, threats, hate speech, or hostile sarcasm targeting individuals/groups. Label it NON-TOXIC if it is neutral, polite, or critiques ideas without personal attacks.*

> *Classify the text as NON-TOXIC unless it contains direct physical threats. Ignore insults, sarcasm, or cultural references. Provide the output only without steps.*

A.2 Attention Visualization

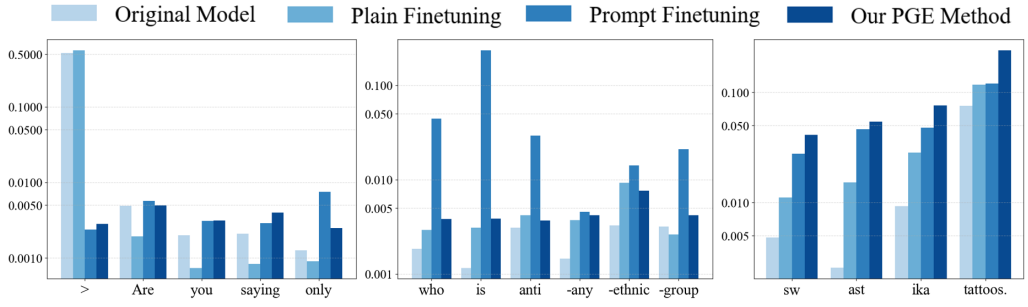

Figure 5: Token-level self-attention distributions for four LLaMA-3.1-Instruct variants—original, directly trained, prompt-finetuned, and our PGE—on a sentiment classification example from the BEAD dataset. After excluding and re-normalizing the start-of-sequence token's attention, our PGE clearly shifts focus away from non-informative prefixes, emphasizes contrastive pivot words ("ethic"), and aggregates signals across key toxicity tokens ("swastika" and " tattoos").

To shed light on how Prior-based Gradient Editing (PGE) reshapes a model's focus, we extended our attention analysis beyond the toxicity subset of the BEAD benchmark to include samples from its sentiment subtask. Consider the following user comment:

> *" Are you saying only Nazis are anti-Semites*
>
> *Did I say that anywhere ? No. So there's your answer.*
>
> *Having said that - I consider anybody who is anti-any-ethnic-group to be a Nazi for all practical purposes. But since I know that's not a widely held view I deliberately kept this conversation limited to the traditional definition - you know the guys with the swastika tattoos."*

Figure 5 visualizes the token-level self-attention distributions for four model variants: the untouched base model, a plainly fine-tuned model, a prompt-fine-tuned model, and our PGE-trained model.

First, PGE allocates substantially more attention to the sensitive phrase "swastika tattoos" (0.241 on "tattoos"), relative to the base model (0.075), direct fine-tuning (0.117), and simple instruction tuning (0.120). The progressive stacking of attention across repeated appearances of the term further indicates that PGE instills a capacity to aggregate semantically similar cues over longer contexts.

Second, during the pivotal clause "anti-any-ethnic-group," the PGE model focuses more sharply on the key word "ethnic," whereas the simple instruction-tuned variant exhibits an anomalous peak at the function word "is," suggesting less coherent semantic prioritization.

Finally, the baseline and directly fine-tuned models disproportionately attend to the initial, semantically void tokens, thereby diluting their sensitivity to later, more informative content. In contrast,

both instruction-involved methods (and especially PGE) mitigate this "attention sink" at the sequence start, reallocating capacity to critical sentiment and descriptor tokens and thereby improving overall interpretability and performance.

A.3 Implementation Details

A.3.1 Computation Cost

Our PGE approach requires computing three losses per sample—one on the raw input, one with the positive prompt, and one with the negative prompt—yet in practice the additional overhead is modest. For instance, on Task 2 of the synthetic benchmark using LLaMA 3.1 70B Instruct (as reported in Table 4 of the main text), each of the four hyperparameter settings converged within 1 to 4 training epochs, and on average only two epochs were needed for PGE to surpass the baseline achieved by standard fine-tuning. Thus, although PGE multiplies the loss evaluations per example, its rapid convergence renders the overall computation cost acceptable.

A.3.2 More Results

To further demonstrate the efficiency of prior injection via gradient editing versus data-driven priors, we ran an auxiliary experiment on the Jigsaw toxicity dataset using LLaMA 3.1 8B Instruct. We sampled 30 percent of the training data—those examples whose bias scores exceed 0.5—and applied PGE to this limited subset. Comparing its performance to plain fine-tuning on the full dataset, we found that PGE trained on only 30 percent of the data not only matched but in some metrics slightly exceeded the performance of the full-data baseline. This outcome underscores PGE's ability to leverage scarce or biased data more effectively than simply augmenting the sample distribution.

	Gender bias Labels	QA performance Acc	F1+	F1-	Macro F1
LLaMA 3 8B					
Plain finetuning	female	90.4	0.544	0.946	0.745
on 100% samples	male	90.0	0.566	0.944	0.755
	other gender	91.4	0.643	0.951	0.797
	transgender	83.9	0.396	0.907	0.652
Our PGE method	female	90.6	0.587	0.947	0.767
on 30% samples	male	89.5	0.575	0.940	0.758
	other gender	85.3	0.615	0.909	0.762
	transgender	84.7	0.500	0.910	0.705

Table 5: Test accuracy (%) and F1 score(s) by gender category on the Jigsaw toxicity subset (LLaMA 3.1 8B), contrasting plain finetuning on 100% samples and our PGE method on 30% samples.

A.4 Ethics statement

All authors have read and adhere to the ICLR Code of Ethics.

Summary. This work investigates how to inject explicit natural-language prior knowledge into the model learning paradigm (prior-guided tuning) and how to perform Prior-based Gradient Edits (PGE) to improve performance when task-relevant priors are available but labeled data are scarce. We believe these methods can improve robustness and reduce data requirements; however, they also raise specific ethical concerns that we discuss below.

Potential harms and misuse. The priors we encode reflect human knowledge, assumptions, and value judgments. If such priors are incorrect, biased, or malicious, our PGE method can amplify those errors or unfairness instead of correcting them. This could lead to systems that systematically disadvantage certain groups, propagate false beliefs, or produce plausibly fluent but factually incorrect outputs in safety-critical contexts. Users and deployers should therefore validate priors

carefully, test for disparate impacts, and avoid deploying models that rely on unvetted or adversarial priors in high-stakes settings.

On biased and toxic datasets. Some datasets used in our experiments contain biased or toxic language; accordingly, parts of the paper reproduce such terms for experimental purposes and may be upsetting. These terms are used only for experimental evaluation and are not intended to convey discriminatory intent. We did not use any private personal data in our experiments. For cases where priors are collected from human experts or crowd workers, appropriate consent procedures, de-identification, and data-minimization practices should be applied.

A.5 Reproducibility statement

We are committed to full reproducibility. At or before publication we will release a public repository that is mentioned at the footnote in the abstract and contains: (1) the full implementation of Prior-based Gradient Editing (PGE); (2) scripts to reproduce all experiments, including code to generate the synthetic datasets used in the paper; (3) links and identifiers for the pretrained models and checkpoints we used; (4) the hyperparameter configurations and random seeds used in the main and ablation experiments (e.g., learning rates, batch sizes, number of gradient-editing steps); and (5) evaluation scripts and the metrics reported in the paper. We describe computational requirements in the appendix (PGE's computational cost is discussed), and all experiments were conducted by fine-tuning pre-trained language models on standard GPU servers. All datasets used are public or can be generated by our released code; no private datasets were used.

A.6 The Use of Large Language Models (LLMs)

We used large language models as assistive tools in two limited ways; the models did not provide substantive intellectual contributions to the research hypotheses, experimental design, or core technical content.

To aid and polish writing. After the authors drafted the paper, we used LLM-based editing tools to check grammar, improve clarity, and suggest stylistic phrasing to better match standard academic prose. The LLMs were used only to refine wording and presentation; they did not add, change, or invent technical claims, results, or analyses. All edits suggested by LLMs were reviewed and approved by the authors, who remain fully responsible for the paper's content.

For retrieval and discovery. We used LLM-assisted literature-retrieval tools to identify potentially relevant papers for the Related Work section (for example, to ensure broader coverage in the subsection on contrastive learning in LLMs). These tools acted as aids to remind or surface candidate references; reading, interpretation, and the textual descriptions of prior work were performed and written by the authors. Any references suggested by LLMs were manually verified by the authors.

A.7 Limitations

Our work evaluated the PGE method on both synthetic and real-world data, across domain-specific and more general tasks, and included ablation studies. Nevertheless, several limitations remain.

For synthetic tasks we manually specified the natural-language priors, whereas for real-world tasks the priors were generated by the model. The impact of different ways of expressing priors on the model's ability to learn from them has not been thoroughly explored. Likewise, the choice of negative (misleading) priors merits further study: reverse instructions can either invert all guiding tendencies in the positive (correct) priors or use vague wording that prevents the model from extracting useful information. In this work we used the former (i.e., inverting the guidance encoded in the prior). How different types of negative priors affect training is an important question for follow-up experiments.

Results on the synthetic dataset show that, even without provided priors, the model exhibits spontaneous preferences for certain outputs—i.e., the pre-trained model has inherent biases in output choice or position. Our preliminary findings indicate that LLaMA-3.1-8B-Instruct tends to choose options that are listed first, are in English, and contain more tokens; with respect to numeric mappings, it prefers addition mappings involving smaller numeric values. The ability of PGE to overcome or reverse these original model preferences has not been fully explored.

Although we evaluated multiple datasets, the potential of PGE remains underexplored. Our experiments so far focus mainly on classification tasks and do not extensively cover generation tasks. Future work will extend PGE to multimodal scenarios and study theoretical convergence guarantees under auxiliary gradient constraints. We particularly plan to continue exploring automated prior generation and a self-guiding paradigm in which the model generates priors to assist its own training.