Bias-Restrained Prefix Representation Finetuning for Mathematical Reasoning

Sirui Liang[1,2,3], **Pengfei Cao**[1,2*], **Jian Zhao**[3,4], **Cong Huang**[3,4], **Jun Zhao**[1,2], **Kang Liu**[1,2*]

[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Zhongguancun Academy, Beijing, China
[4]Zhongguancun Institute of Artificial Intelligence, Beijing, China
liangsirui2024@ia.ac.cn, {pengfei.cao, jzhao, kliu}@nlpr.ia.ac.cn, {jianzhao, huangcong}@zgci.ac.cn

Abstract

Parameter-Efficient finetuning (PEFT) enhances model performance on downstream tasks by updating a minimal subset of parameters. Representation finetuning (ReFT) methods further improve efficiency by freezing model weights and optimizing internal representations with fewer parameters than PEFT, outperforming PEFT on several tasks. However, ReFT exhibits a significant performance decline on mathematical reasoning tasks. To address this problem, the paper demonstrates that ReFT's poor performance on mathematical tasks primarily stems from its struggle to generate effective reasoning prefixes during the early inference phase. Moreover, ReFT disturbs the numerical encoding and the error accumulats during the CoT stage. Based on these observations, this paper proposes **Bias-REstrained Prefix Representation FineTuning (BREP ReFT)**, which enhances ReFT's mathematical reasoning capability by truncating training data to optimize the generation of initial reasoning prefixes, intervening on the early inference stage to prevent error accumulation, and constraining the intervention vectors' magnitude to avoid disturbing numerical encoding. Extensive experiments across diverse model architectures demonstrate BREP's superior effectiveness, efficiency, and robust generalization capability, outperforming both standard ReFT and weight-based PEFT methods on the task of mathematical reasoning.

Code — https://github.com/LiangThree/BREP

Introduction

Large language models (LLMs) exhibit remarkable performance across a wide range of tasks (Achiam et al. 2023; Anil et al. 2023). Task-specific finetuning further enhances LLMs' capabilities (Dai and Le 2015). However, full parameter finetuning consumes substantial computational costs (Brown et al. 2020; Houlsby et al. 2019). Parameter-Efficient finetuing (PEFT) methods (Asai et al. 2022; He et al. 2021), such as Low-Rank Adaptation (LoRA) (Hu et al. 2022), Prefix-tuning (Li and Liang 2021) and Adapter (Houlsby et al. 2019), mitigate this issue by updating a subset of full parameters, yet still require a significant number of trainable parameters (Ding et al. 2023, 2022; Aghajanyan, Zettlemoyer, and Gupta 2020). A recently proposed finetuning paradigm, Representation Finetuning (ReFT), modifies the internal representations of the model without adjusting model parameters, and the model output could be intervened and modified targetly. In this way, greater efficiency could be achieved than PEFT (Li et al. 2023; Wu et al. 2024b,a).

Actually, there are several adaptation paradigms for ReFT, including per-layer bias injection (Li et al. 2023), scaling and biasing operations on layer-wise representations (Wu et al. 2024a), and representation interventions within linear subspaces spanned by low-rank projection matrices (Wu et al. 2024b). This work focuses on the most generalizable and widely applicable ReFT approach: applying **learnable scaling and biasing operations** to the model's intermediate representations. Other ReFT methodologies can be conceptualized as specific instances or variations of this fundamental operation(Li et al. 2023; Wu et al. 2024b).

ReFT demonstrates superior parameter efficiency and achieves better performance in commonsense reasoning, instruction-following (Li et al. 2023; Wang et al. 2025) but exhibits a significant performance gap in mathematical reasoning compared to LoRA, averaging 11.5% in the GSM8K dataset. Wu et al. (2024b) indicated that ReFTs may have trouble in CoT reasoning than single-step commonsense reasoning tasks (Wei et al. 2022; Cobbe et al. 2021; Hendrycks et al. 2021). Some representation engineering methods explored intervention techniques to enhance model reasoning ability (Tang et al. 2025; Højer, Jarvis, and Heinrich 2025). However, the improvement achieved by these methods is small compared to the weight-based PEFT strategies.

Our diagnostic analysis reveals two primary factors for this failure (as shown in Figure 1): **1) Misleading reasoning prefixes**: ReFT struggles to generate effective initial reasoning steps (*the first k tokens generated by model when answering questions*), which are critical for guiding the model towards a correct solution path (Ji et al. 2025; Qi et al. 2025). **2) Disturbing numerical encoding**: The intervention vectors used by ReFT can disrupt the model's internal representation of numbers, which is a fundamental component of mathematical reasoning. During the autoregressive generation of CoT, this error continues to accumulate as the output length increases, directly leading to the calculation error (Cao et al. 2024; Zhang, Tang, and Hao 2025).

*Corresponding author.

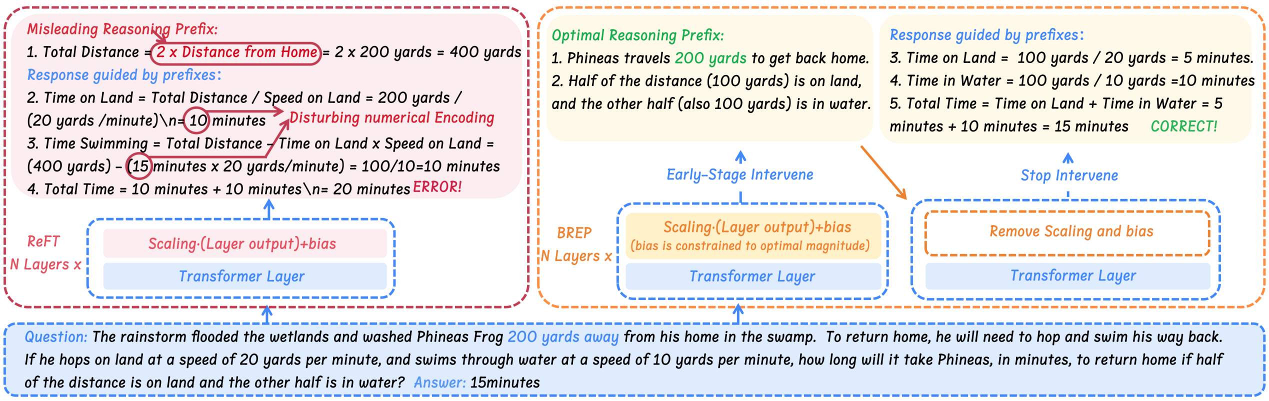

Figure 1: The overview of ReFT and BREP. An example of misleading reasoning prefix and disturbing numerical encoding are shown in ReFT.

Based on the aforementioned observations, this work proposes **Bias-Restrained Prefix Representation Finetuning (BREP)**. It is a ReFT method that enhances mathematical reasoning performance through two components: **1) Prefix Training and Early-Stage Intervention**: Training data is truncated to focus on initial mathematical reasoning steps, learning high-quality prefix representations. During inference, applying intervention only to the first k tokens, which maximizes the benefits of prefix training and prevents bias accumulation in CoT stage. **2) Bias Constraint Training**: Excessively large-magnitude intervene vectors damages the model's numerical encoding, while too short intervene vectors results in ineffective utilization of ReFT's enhancements. This paper leverages Proportional-Integral-Derivative (PID) (Åström and Hägglund 2006) control during the train process, enabling intervene vectors constrained within an optimal magnitude. BREP significantly improves the performance of the model in mathematical reasoning tasks surpassing existing ReFT and PEFT methods while demonstrating robust generalization capabilities. The contributions of our work are summarized as follows:

- Our analysis reveals that the primary causes of ReFT's underperformance in mathematical reasoning tasks are misleading initialization of reasoning prefix and excessively large-magnitude intervene vector disturbing numerical encoding.
- The paper proposes bias-restrained prefix ReFT method that enhances mathematical reasoning ability by optimizing the reasoning prefix and regulating intervention vector magnitudes to avoid numerical encoding error.
- A comprehensive experiment is conducted to demonstrate the effectiveness, efficiency, and robust generalization of our method.

ReFT Definition

In this section, a definition of ReFT is provided. For an input token sequence $\boldsymbol{x} = (x_1, \ldots, x_n)$ tokens are first embedded into vectors $\boldsymbol{h}^0 = (\boldsymbol{h}_1^0, \ldots, \boldsymbol{h}_n^0)$ where each $\boldsymbol{h}_i^0 \in \boldsymbol{R}^d$. They are processed by l transformer layers. Each layer j $(1 \leq j \leq l)$ computes:

$$\boldsymbol{h}^j = \boldsymbol{h}^{j-1} + \mathbf{A}^j + \mathbf{F}^j. \tag{1}$$

with $\mathbf{A}^j$ and $\mathbf{F}^j$ being multi-head self-attention and feed-forward network outputs respectively.

A widely used ReFT adjustment (Wu et al. 2024a) method can be formalized as:

$$\boldsymbol{h}^j = \mathbf{W} \odot \left(\boldsymbol{h}^{j-1} + \mathbf{A}^j + \mathbf{F}^j\right) + \boldsymbol{b}, \tag{2}$$

where $\mathbf{W}, \boldsymbol{b} \in \boldsymbol{R}^d$ are learnable parameters, $\odot$ denotes element-wise multiplication (Hadamard product), and $\boldsymbol{h}^j$ is the ReFT modified representation. These vectors directly guide the model output and its magnitude and direction directly determine the effectiveness of the ReFT.

Problem Diagnosis

In this section, interpretable analyze experiments are conducted to reveal critical problems within ReFT in mathematical reasoning tasks.

Misleading Reasoning Prefixes

ReFT always fails to generate optimal reasoning steps during the early stage of mathematical problem solving. To validate this, prefixes of varying lengths were generated from a ReFT-finetuned model on 100 GSM8K (Cobbe et al. 2021) cases. Under a temperature setting of 0.6, the base model continues to generate 10 responses along the given reasoning prefix. The answer results of the base model without prefix are also provided for comparison. As shown in Figure 2, shorter ReFT prefixes marginally underperform the base model without prefix, and the performance gap becomes more pronounced with longer ReFT prefixes. These observations suggest a suboptimal effectiveness of ReFT in early stage of mathematical reasoning. It is necessary to enhance the capability of ReFT in the reasoning build-stage and resolve the critical performance decline caused by longer representation intervention (Xu et al. 2025).

Disturbing Numerical Encoding

Theoretical Analysis Current research provides evidence that LLMs encode the value of number linearly (Zhu, Dai,

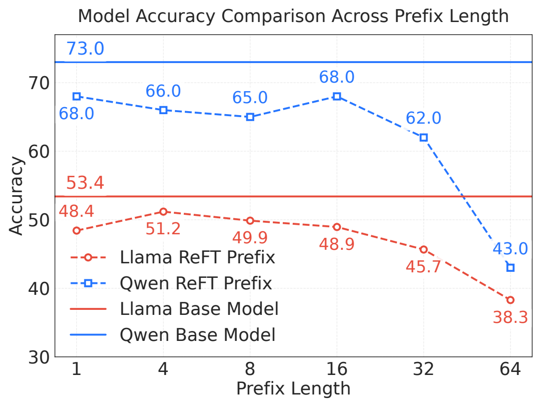

Figure 2: The impact of ReFT and Base prefix with various prefix length on mathematical reasoning performance.

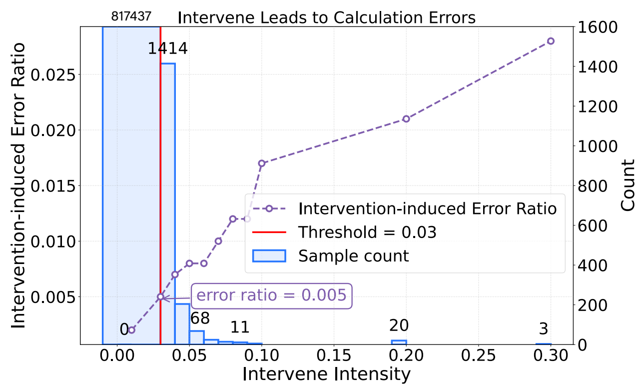

Figure 3: Effect of interventions on numerical encoding. X-axis: The intensity of intervention along the direction of numerical coding. Line plot (left y-axis): Error probability of four-digit addition under interventions. Bar plot (right y-axis): Intervention intensity of ReFT projected onto the number encoding direction.

and Sui 2025). Formally, given a set of hidden states $\mathcal{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$ and their corresponding original numbers $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, a linear regressor P can be trained to predict the number encoded in the internal representation of LLM:

$$P = \mathbf{N}\boldsymbol{h} + \boldsymbol{d}, \tag{3}$$

where $\mathbf{N} \in \boldsymbol{R}^d$ and $\boldsymbol{d}$ are the weight parameters of P.

It is reasonable to hypothesize that, when ReFT modifies the hidden states, it may have an impact on digital encoding. Considering that ReFT induces minimal adjustments to weight parameter $\mathbf{W}$ during mathematical task training, this work focuses on analyzing the influence of bias parameter $\boldsymbol{b}$. Define a linear probe parameterized by $(\mathbf{N}, \boldsymbol{d})$. When ReFT introduces a bias intervention $\boldsymbol{\alpha}$ at a particular layer during inference, the perturbed numerical encoding becomes:

$$\hat{P} = \mathbf{N}(\boldsymbol{h} + \boldsymbol{\alpha}) + \boldsymbol{d} = \mathbf{N}\boldsymbol{h} + \boldsymbol{d} + \mathbf{N}\boldsymbol{\alpha} = P + \mathbf{N}\boldsymbol{\alpha}, \tag{4}$$

where P denotes the original probe prediction. When $\boldsymbol{\alpha}$ only affects the number in the direction of the digital encoding, $\boldsymbol{\alpha}$ is mapped to $\mathbf{N}$:

$$\hat{P} \approx P + \mathbf{N}(\mathbf{N} \cdot \boldsymbol{\alpha}) = P + c\|\mathbf{N}\|^2, \tag{5}$$

where c denotes the projection coefficient of α onto N. Generalizing to a complete forward pass through L layers:

$$\hat{P}_{cum} = P + \left(\sum_{i=1}^{L} c_i\right) \|\mathbf{N}\|^2. \tag{6}$$

As token generated during autoregressive inference, **this systematic error induced by ReFT progressively accumulates during CoT outputs**.

Empirical Validation To validate our analysis, this paper first quantifies the intervention intensity threshold that disrupts the computational capacity of the model. During four-digit addition computations, **positive** interventions of magnitude δ along the numerical encoding direction ($\hat{n} = \frac{N}{\|N\|}$) are applied to the model. An intervention-induced error is defined as the calculation result after intervention is changed:

$$\text{Model}_{\text{positive-intervention}}(\delta) > \text{Model}_{\text{pre-intervention}}, \tag{7}$$

$\text{Model}_{\text{positive-intervention}}$ adds positive intervene to the model which increases the calculation result while negative intervene decreases it (model calculates 1234+4321=5560 positive-intervention $>$ 5555 pre-intervention). As shown in Figure 3 (purple curve), as the intensity of interference increases, the error rate increases. This work establishes $\delta = 0.04$ as the critical threshold, interventions beyond this magnitude induce >5 computational errors per 1,000 arithmetic problems.

Subsequently, the intervention intensity of ReFT vectors in the numerical encoding direction is measured. The hidden state deviation α in equation (4) is projected before and after ReFT onto the numerical encoding direction $\hat{n}$ (this quantifies the equivalent intervention intensity along the numerical encoding axis for bias introduced by ReFT). As illustrated in Figure 3 (blue bar), more than 1,000 ReFT intervene vectors exceed the critical threshold ($\delta_{\text{crit}} = 0.04$). This statistically significant deviation directly explains 21 identified errors related to numbers (consist of confusing and forgetting key data that account for 18.4% of all errors), empirically confirming that ReFT perturbs numerical encoding. The detailed numerical probe training process and comprehensive error statistics are documented in the Appendix.

Methodology

Based on the aforementioned observations, this paper proposes **Bias-REstrained Prefix Representation Finetuning (BREP ReFT)**, which comprises two core components: 1) Prefix Training and Early-Stage Intervention, 2) Bias Constraint Training.

Prefix Training and Early-Stage Intervention

Prefix Training Considering that the reasoning supervision signal is concentrated in early tokens (Qi et al. 2025) and ReFT models always generate misleading reasoning prefixes, BREP **truncates the dataset and only utilizes the**

first k tokens of each response sequence for training. The per-token supervision signal between the prefix training and full training is:

$$R_t(i) = \frac{1}{l_p} logp_t(y_t|x, y_{1:t-1}) - \frac{1}{l_f} logp_t(y_t|x, y_{1:t-1}), \quad (8)$$

where p_t denotes the conditional probability distribution generated by the model at position t, x represents question, y_t is the target token. l_p denotes the truncation length and l_f is the complete sequence length. It can be observed that prefix training can provide a stronger signal strength when $t \le l_p < l_f$.

Early-Stage Intervention During inference, BREP **applies interventions only to the first n tokens** ($n \le l_p$). The optimization target becomes the prefix cumulative reward:

$$R_{cum} = \sum_{t=1}^{n} R_t(i). \quad (9)$$

This enables full use of the prefix training supervision signal and avoids the accumulation of bias caused by continuous intervention. Furthermore, the optimized gradient directionality is as follows:

$$\nabla_\theta R_{cum}^{intervene} \propto \sum_{t=1}^{n} \nabla_\theta logp(y_t \mid x, y_{1:t-1}), \quad (10)$$

which formulates an ideal target gradient which is derived exclusively from the first n tokens of the sequence, $\nabla_\theta R$ represents the direction of the cumulative gradient of the training.

$$\nabla_\theta R_{cum}^{prefix} \propto \nabla_\theta R_{cum}^{intervene} + \underbrace{\sum_{t=n+1}^{l_p} \alpha_t \nabla_\theta logp(y_t \mid \cdot)}_{\textbf{contamination signal}}, \quad (11)$$

$$\nabla_\theta R_{cum}^{full} \propto \nabla_\theta R_{cum}^{intervene} + \underbrace{\sum_{t=n+1}^{l_f} \alpha_t \nabla_\theta logp(y_t \mid \cdot)}_{\textbf{contamination signal}}, \quad (12)$$

describes the gradient for prefix training and full training. Prefix training focuses on optimization of initial reasoning, while full training suffers significant contamination beyond position n.

In general, prefix truncation brings a stronger supervised signal and a gradient direction that is focused on early-stage reasoning thoughts. Early-stage intervention makes full use of these signals while avoiding the accumulation of errors caused by intervention.

Bias Constraint Training

To constrain the magnitude of the learned intervene vector, this paper develops a generalized framework for precision-controlled bias magnitude during training. The total loss function $\mathcal{L}_{\text{total}}$ consists of two components: a cross-entropy loss term and an adaptive bias magnitude adjustment.

Cross-Entropy Loss Given target sequence $Y = \{y_1, y_2, \ldots, y_k\}$ containing the first k response tokens, the cross-entropy loss at timestep t is computed as:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{k} \sum_{t=1}^{k} logp_t(y_t \mid x, y_{1:t-1}), \quad (13)$$

where $p_t^{y_t} \in [0, 1]$ denotes the softmax-normalized probability of the target token y_t at position t, x represents question and V represents the vocabulary size ($\sum_{i=1}^{V} p_t^i = 1$).

Adaptive Bias Magnitude Adjustment To accelerate early-stage learning while avoiding excessive bias in learning, an adaptive weight $w(t)$ is regulated via Proportional-Integral-Derivative (PID) control (Åström and Hägglund 2006; Tohma et al. 2025) which represents a foundational control algorithm in dynamical systems. The weight update mechanism follows:

$$b(t) = \frac{1}{L} \sum_{l=1}^{L} \|\mathbf{b}_l(t)\|_2, \quad (14)$$

where $b(t)$ denotes the mean bias vector learned across all layers at the current training step.

$$e(t) = b_{\text{target}} - b(t), \quad (15)$$

where b_{target} is the target bias magnitude (b_{target} *will be analyzed in subsection Optimal Bias Magnitude*) and $e(t)$ is the distance from the target bias magnitude.

$$\Delta w(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t)dt + K_d \cdot \frac{de(t)}{dt}, \quad (16)$$

where K_p provides instantaneous error correction, K_i eliminates steady-state offset through accumulated error integration, and K_d enhances stability ($K_p = 1 \times 10^{-1}, K_i = 1 \times 10^{-4}, K_d = 1 \times 10^{-2}$). Our method automatically constrain the bias magnitude through $\Delta w(t)$.

$$w(t+1) = clip(w(t) \cdot (1 + \alpha \cdot \Delta w(t)), w_{min}, w_{max}), \quad (17)$$

α is a smoothing factor to prevent excessive fluctuations in the loss ($\alpha = 5$) and w_{max}, w_{min} ensure stable update range($w_{max} = 1 \times 10^{-1}, w_{min} = 1 \times 10^{-5}$).

Bias constrained training enables precise optimization of bias magnitude acquisition, ensuring models converge to empirically validated optimal magnitude. The total loss is formulated as:

$$\mathcal{L}_{\text{total}} = w(t) \cdot \mathcal{L}_{\text{ce}}. \quad (18)$$

This enables the ReFT intervention vector to learn an optimal magnitude, ensuring its effectiveness while preventing disturbing the numerical encoding.

Experiments

Experimental Setup

Datasets. The experiments utilize two series of open-source LLMs (Llama (Grattafiori et al. 2024) and Qwen (Yang et al. 2025)) on mathematical reasoning tasks. Simple mathematical reasoning tasks are finetuned on a combination of arithmetic reasoning datasets MATH10K (Hu

Model	Method	Math10K				PRM800K		
		GSM8K	SVAMP	MathQA	Avg	MATH500	AMC23	Avg
Llama3-8B -Instruct	Base	80.0	88.9	**55.0**	74.6	40.4	**57.5**	**49.0**
	LoRA	81.1 ↑1.1	**90.0** ↑1.1	54.0 ↓1.0	75.0 ↑0.4	39.3 ↓1.1	53.8 ↓3.7	46.6 ↓2.4
	RED	73.8 ↓6.2	88.9 =	51.3 ↓3.7	71.3 ↓3.3	41.5 ↑1.1	56.4 ↓1.1	**49.0** =
	LoReFT	78.8 ↓1.2	80.7 ↓8.2	44.7 ↓10.3	68.1 ↓6.5	37.0 ↓3.4	35.0 ↓22.5	36.0 ↓13.0
	BREP(ours)	**82.8** ↑2.8	89.5 ↑0.6	54.3 ↓0.7	**75.5** ↑0.9	**42.8** ↑2.4	52.5 ↓5.0	47.7 ↓1.3
Llama3.1-8B -Instruct	Base	87.3	94.4	72.8	84.8	59.8	**77.5**	68.7
	LoRA	87.9 ↑0.6	94.8 ↑0.4	72.9 ↑0.1	85.2 ↑0.4	43.7 ↓16.1	61.5 ↓16.0	52.6 ↓16.1
	RED	82.9 ↓4.4	91.1 ↓3.3	67.5 ↓5.3	80.5 ↓4.3	58.1 ↓1.7	70.0 ↓7.5	64.1 ↓4.6
	LoReFT	84.1 ↓3.2	89.1 ↓5.3	57.0 ↓15.8	76.7 ↓8.1	42.6 ↓17.2	42.5 ↓35.0	42.6 ↓26.1
	BREP(ours)	**91.6** ↑4.3	**96.8** ↑2.4	**73.3** ↑0.5	**87.2** ↑2.4	**63.5** ↑3.7	**77.5** =	**70.5** ↑1.8
Qwen2.5-Math -7B-Instruct	Base	95.4	96.1	82.7	91.4	81.2	60.0	70.6
	LoRA	96.4 ↑1.0	**96.5** ↑0.4	83.9 ↑1.2	92.3 ↑0.9	80.8 ↓0.4	**74.4** ↑14.4	77.6 ↑7.0
	RED	95.7 ↑0.3	95.9 ↓0.2	81.7 ↓1.0	91.1 ↓0.3	78.2 ↓3.0	66.7 ↑6.7	72.5 ↑1.9
	LoReFT	93.6 ↓1.8	95.3 ↓0.8	74.5 ↓8.2	87.8 ↓3.6	78.8 ↓2.4	72.5 ↑12.5	75.7 ↑5.1
	BREP(ours)	**96.9** ↑1.5	96.2 ↑0.1	**84.3** ↑1.6	**92.5** ↑1.1	**82.0** ↑0.8	**74.4** ↑14.4	**78.2** ↑7.6
Qwen3-8B	Base	95.1	96.7	**86.5**	92.8	82.0	85.0	83.5
	LoRA	95.1 =	96.8 ↑0.1	86.2 ↓0.3	92.7 ↓0.1	81.8 ↓0.2	**87.5** ↑2.5	84.7 ↑1.2
	RED	87.9 ↓7.2	91.8 ↓4.9	77.3 ↓9.2	85.7 ↓7.1	54.2 ↓27.8	35.0 ↓50.0	44.6 ↓38.9
	LoReFT	87.1 ↓8.0	96.3 ↓0.4	72.8 ↓13.7	85.4 ↓7.4	72.4 ↓9.6	80.0 ↓5.0	76.2 ↓7.3
	BREP(ours)	**95.3** ↑0.2	**97.4** ↑0.7	86.3 ↓0.2	**93.0** ↑0.2	**82.6** ↑0.6	**87.5** ↑2.5	**85.1** ↑1.6
Qwen3-14B	Base	96.5	96.3	87.4	93.4	84.2	82.5	83.4
	LoRA	96.6 ↑0.1	96.1 ↓0.2	87.1 ↓0.3	93.3 ↓0.1	84.4 ↑0.2	70.0 ↓12.5	77.2 ↓6.2
	RED	93.5 ↓3.0	95.8 ↓0.5	82.9 ↓4.5	90.7 ↓2.7	63.2 ↓21.0	57.5 ↓25.0	60.4 ↓23.0
	LoReFT	88.5 ↓8.0	**96.8** ↑0.5	74.2 ↓13.2	86.5 ↓6.9	73.0 ↓11.2	77.5 ↓5.0	75.3 ↓8.1
	BREP(ours)	**96.8** ↑0.3	96.2 ↓0.1	**87.6** ↑0.2	**93.5** ↑0.1	**84.6** ↑0.4	**90.0** ↑7.5	**87.3** ↑3.9

Table 1: Performance comparison of Base (the original model without finetuning), RED (Wu et al. 2024a), LoRA (Hu et al. 2022), LoReFT (Wu et al. 2024b) and BREP(ours). Up arrows (↑) indicate improvement over base model, down arrows (↓) indicate decrease. Best method in each group marked in **bold**.

et al. 2023) and tested on standard benchmarks including GSM8K (Cobbe et al. 2021), SVAMP (Patel, Bhattamishra, and Goyal 2021) and MATHQA (Amini et al. 2019). Complex mathematical reasoning tasks are finetuned on PRM800K (Lightman et al. 2023) which includes step-level human feedback labels and test on competitive benchmarks (MATH500 (Hendrycks et al. 2021) and AMC).

Baselines. Different finetuning methods are compared, such as the ReFT method (RED (Wu et al. 2024a) and LoReFT (Wu et al. 2024b)), weight-based PEFT methods (LoRA (Hu et al. 2022)), with Zero-shot CoT.

Implementation Details. Our experiments use two series open-source LLMs: Llama3-8B-Instruct, Llama3.1-8B-Instruct (Grattafiori et al. 2024), Qwen2.5-Math-7B-Instruct (Bai et al. 2023), Qwen3-8B (Yang et al. 2025) and Qwen3-14B. Each method was fine-tuned using 5,000 data. All of our experiments are run with a single Nvidia A100 80G GPU. More experimental details are provided in Appendix.

Experimental Results

To verify the effectiveness of our method in mathematical reasoning tasks, we compare it with the ReFT and weight-based PEFT methods. As shown in Table 1. Vanilla ReFT methods (*e.g.*, RED) and its variant LoReFT achieve performance marginally below the base model on mathematical reasoning tasks, while LoRA finetuning improves accuracy on most benchmarks. In contrast, BREP substantially enhances mathematical reasoning in nearly all models and test sets. Crucially, comparative analysis against weight-based PEFT methods reveals BREP's robustness: although LoRA achieves competitive results on simpler math reasoning tasks, it exhibits unstable performance on complex benchmarks (*e.g.*, MATH500). BREP maintains strong performance across difficulty levels, demonstrating notable gains in challenging problems. This indicates that BREP's constraint representation learning enables stronger generalization for multi-step mathematical reasoning.

Ablation Study

Ablation studies are conducted by systematically removing individual modules. Table 2 demonstrates the critical contri-

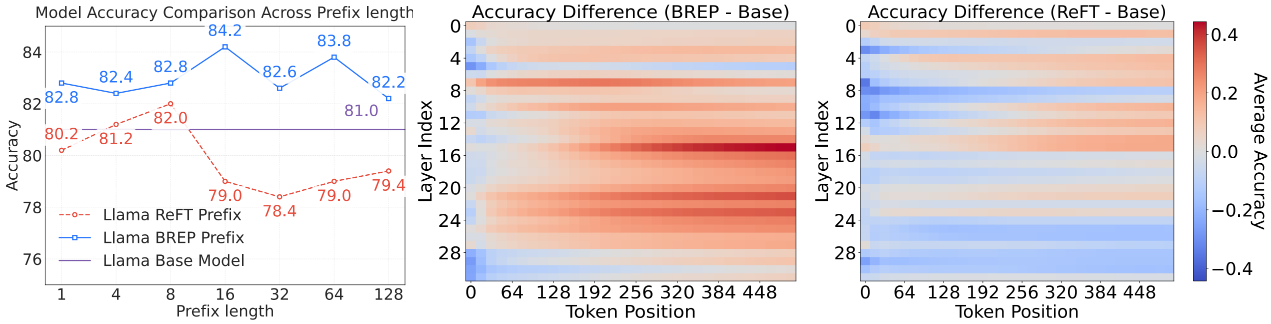

Figure 4: The analysis of BREP. (a) Comparison of mathematical reasoning effectiveness guided by BREP prefixes and ReFT prefixes. (b) Numerical faithfulness performance gap between BREP and Base model. (c) Numerical faithfulness performance gap between ReFT and Base model. (Red indicates faithfulness improvement, blue indicates faithfulness degradation.)

Model	Ablation	GSM8K	MATH500
Llama3-8B -Instruct	BREP	82.8	42.7
	w/o PT	81.0 $\downarrow$1.8	40.2 $\downarrow$2.5
	w/o BCT	80.0 $\downarrow$2.8	39.4 $\downarrow$3.3
	w/o ESI	80.4 $\downarrow$2.4	37.6 $\downarrow$5.1
Qwen3-8B	BREP	96.9	82.0
	w/o PT	95.5 $\downarrow$1.4	79.4 $\downarrow$2.6
	w/o BCT	94.9 $\downarrow$2.0	79.8 $\downarrow$2.2
	w/o ESI	95.1 $\downarrow$1.8	81.6 $\downarrow$0.4

Table 2: Ablation study of BREP, "w/o" denotes the removal of subcomponent. PT: Prefix Truncation, BCT: Bias Constraint Training, ESI: Early-Stage Intervention.

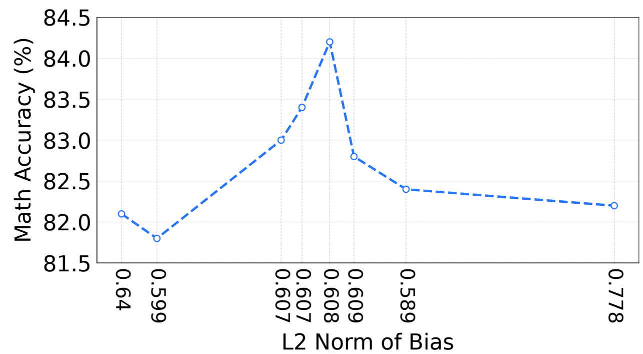

Figure 5: The relationship between model performance and bias magnitude.

butions of three core components—Prefix Truncation (PT), Bias Constraint Training (BCT), and Early-Stage Intervention (ESI). The most pronounced decline observed when removing Bias Constraint Training, particularly for complex reasoning, validating its necessity in ensure accurate calculation through intervention vector regulation. Early-Stage Intervention removal harms performance similarly, this aligns with our hypothesis that interventions within longer reasoning chains enlarge error accumulation. Prefix Truncation's absence yields smaller declines. Notably, performance gaps widen with reasoning complexity: average drops on MATH500 ($\downarrow$2.68%) exceed those on GSM8K ($\downarrow$2.03%), underscoring BREP's enhanced efficacy for multi-step long CoT problems. These results establish the synergistic necessity of all components for BREP.

BREP Exhibits Superior Reasoning Prefix

To verify whether BREP effectively addresses ReFT's poor performance in generating reasoning prefixes during the early stage (analyzed in *Misleading Reasoning Prefixes*), 500 problems were randomly selected from GSM8K and used both BREP and ReFT models to generate prefixes of varying lengths. The base model then continued answering based on the given problems and prefixes. As shown in Figure 4(a), BREP consistently outperforms ReFT by +0.8–5.2% accuracy across different prefix lengths, with its advantage becoming more pronounced as the prefix length increases. This demonstrates that BREP generates significantly higher-quality initial reasoning paths. In complex, multi-step reasoning scenarios, BREP can consistently enlarge the optimaztion gap with longer prefixes.

BREP Encodes Numbers Faithfully

This section verifies whether BREP effectively mitigates the issue of ReFT in disturbing numerical encoding (analyzed in *Deviation accumulation in CoT*). Probes are trained to detect digital faithfulness within the model (Li et al. 2023; Belinkov 2022), the detailed training data construction method and probe accuracy are introduced in *Probe Details* in Appendix. Figure 4(b) (*BREP-Base*) shows the accuracy gap between BREP and the Base model across all layers and token positions. Conversely, Figure 4(c) (*ReFT-Base*) illustrates the accuracy gap between ReFT and the Base model. It can be observed visually that BREP exhibited significantly higher numerical faithfulness compared to the Base model, while ReFT performed below the Base model's faithfulness level. Detailed data statistics are available in Appendix. This demonstrates the efficacy of BREP in resolving the disturbation of ReFT on numerical encoding.

Model	Method	BoolQ	PIQA	GPQA
Llama3 -8B -Instruct	**Base**	19.4	42.4	27.0
	LoRA	18.3 ↓1.1	47.2 ↑4.8	21.9 ↓5.1
	RED	15.0 ↓4.4	63.9 ↑21.5	24.6 ↓2.4
	BREP(ours)	**20.7** ↑1.3	**60.7** ↑18.3	**28.2** ↑1.2
Qwen3 -8B	**Base**	68.9	40.1	48.7
	LoRA	69.3 ↑0.4	40.6 ↑0.5	48.3 ↓0.4
	RED	70.1 ↑1.2	23.0 ↓17.1	49.9 ↑1.2
	BREP(ours)	**69.1** ↑0.2	**42.3** ↑2.2	**50.2** ↑1.5

Table 3: Comparison of generalization performance.

BREP Achieves Optimal Bias Magnitude

The relationship between ℓ_2-norm of bias vectors ($\|\mathbf{bias}\|_2$) and model performance is analyzed in this section. As Figure 5 illustrates, optimal performance occurs when the ℓ_2-norm of learned bias vectors converges to a critical threshold of **0.608**, confirming a strong correlation between bias size and model performance. An excessively small ℓ_2-norm indicates insufficient influence of the bias term, suggesting underutilization of the ReFT mechanism. Conversely, an excessively large ℓ_2-norm causes disturbing numerical encoding and error accumulation. Standard ReFT exceeds $\|bias_{\text{ReFT}}\|_2 = 1.07$, resulting a 6.2% performance decline in GSM8K. Furthermore, significant variations in optimal vector magnitude are observed in both different model families and differently-sized models within the same family. Notably, BREP employs **the same target bias within a series of LLM** ($target_{\text{Llama}}$=1.0; $target_{\text{Qwen}}$=1.5), while our PID controller dynamically optimizes $\|bias\|_2$, contributing to the robustness during training. Optimal bias statistical data and train trend are shown in Appendix.

BREP Has Stronger Generalization Cabilities

Generalization Capability. BREP's generalization capability was further verified by evaluating models trained on mathematical datasets on out-of-domain commonsense tasks (BoolQ (Clark et al. 2019) and PIQA (Bisk et al. 2020)) and specialized science benchmarks GPQA (Rein et al. 2024). As shown in Table 3, BREP achieves the most significant cross-task generalization, delivering the strongest average performance gains across all model architectures. LoRA shows slight improvements ($< 1.9\%$) on commonsense tasks but suffers catastrophic degradation on GPQA. RED exhibits high instability on two tasks. Although both BREP and RED enhance model performance through representation manipulation, there are significant differences in their generalization abilities. This phenomenon may stem from smaller vector magnitudes map representations into a broader task subspace, which enhances the model's reasoning and thinking capabilities rather than optimizing task-specific performance.

Related Work

Representation Engineering. Representation Engineering emerged as a paradigm to enhancing the performance, controllability, and efficiency of LLMs by directly manipulating their internal representations (Mikolov, Yih, and Zweig 2013; Liu et al. 2023b; Singh et al. 2024; Park, Choe, and Veitch 2023; Nanda, Lee, and Wattenberg 2023). Unlike traditional weight-based PEFT methods (**?**Liu et al. 2023a; Zhang et al. 2023, 2024), ReFT treats representations (*e.g.*, hidden states) as fundamental units of analysis and control. Current research focuses on two directions. **Representation Editing** such as RED (Wu et al. 2024a) modify representations to reduce trainable parameters while maintaining performance on several tasks. LoReFT (Wu et al. 2024b) extend this via low-rank subspace interventions, achieving greater parameter efficiency. CRFT (Huang et al. 2025) further identifies and optimizes critical representations. **Cognitive Phenomenon Modeling** like RepE (Zou et al. 2024) enables monitoring and steering high-level behaviors (*e.g.*, truthfulness, emotions) by manipulating vectors. Honesty vectors are used to detect hallucinations (Li et al. 2023). Overall, representation engineering offers more lightweight and interpretable methods for enhancing the performance of LLM by directly manipulating their internal representations.

Prefix Finetuning. Recent studies reveal that the initial tokens in LLM responses carry critical signals that influence model behavior (Lin et al. 2023; Kumar et al. 2025). Qi et al. (2025) define shallow preference signals, demonstrating that training on truncated responses achieves competitive or superior reward modeling and DPO performance, suggesting alignment methods often prioritize early tokens over holistic response quality. Ji et al. (2025) extend this insight to reasoning tasks, confirming that early tokens anchor the model's problem-solving approach, finetuning exclusively on prefixes preserves reasoning efficacy while reducing computational costs by 75-99%. These works establish that prefix signals govern preference learning and reasoning in LLMs, advocating for more efficient and effective alignment strategies.

Conclusion

This work addresses the significant performance degradation of ReFT on mathematical tasks. Our analysis reveals misleading reasoning prefixes and disturbing numerical encoding are two primary causes for this problem. Based on this, this paper proposes Bias-Restrained Prefix Representation Finetuning, a novel framework that achieves superior performance to both ReFT and weight-based PEFT methods across multiple mathematical benchmarks while maintaining strong generalization capabilities on several tasks. This work highlights the potential for implementing lighter, more parameter-efficient, and more interpretable ReFT approaches in mathematical reasoning tasks.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (L243006) and the National Natural Science Foundation of China (No.62406321). This work was also supported by the CCF-NetEase ThunderFire Innovation Research Funding (NO.202507).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aghajanyan, A.; Zettlemoyer, L.; and Gupta, S. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.

Amini, A.; Gabriel, S.; Lin, P.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Anil, R.; Dai, A. M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Asai, A.; Salehi, M.; Peters, M. E.; and Hajishirzi, H. 2022. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. *arXiv preprint arXiv:2205.11961*.

Åström, K. J.; and Hägglund, T. 2006. *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Belinkov, Y. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*.

Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI conference on artificial intelligence (AAAI)*, volume 34, 7432–7439.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 1877–1901.

Cao, B.; Lin, H.; Han, X.; and Sun, L. 2024. The life cycle of knowledge in big language models: A survey. *Machine Intelligence Research*, 21(2): 217–238.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dai, A. M.; and Le, Q. V. 2015. Semi-supervised sequence learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 28.

Ding, N.; Lv, X.; Wang, Q.; Chen, Y.; Zhou, B.; Liu, Z.; and Sun, M. 2023. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; and Neubig, G. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Højer, B.; Jarvis, O.; and Heinrich, S. 2025. Improving reasoning performance in large language models via representation engineering. *arXiv preprint arXiv:2504.19483*.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*.

Hu, Z.; Wang, L.; Lan, Y.; Xu, W.; Lim, E.-P.; Bing, L.; Xu, X.; Poria, S.; and Lee, R. K.-W. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Huang, C.; Yan, S.; Xie, L.; Lin, B.; Fan, S.; Xin, Y.; Cai, D.; Shen, C.; and Ye, J. 2025. Enhancing Chain-of-Thought Reasoning with Critical Representation Fine-tuning. *arXiv preprint arXiv:2507.10085*.

Ji, K.; Xu, J.; Liang, T.; Liu, Q.; He, Z.; Chen, X.; Liu, X.; Wang, Z.; Chen, J.; Wang, B.; et al. 2025. The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models. *arXiv preprint arXiv:2503.02875*.

Kumar, A.; He, Y.; Markosyan, A. H.; Chern, B.; and Arrieta-Ibarra, I. 2025. Detecting Prefix Bias in LLM-based Reward Models. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, 3196–3206.

Li, K.; Patel, O.; Viégas, F.; Pfister, H.; and Wattenberg, M. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems (NeurIPS)*.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*.

Lin, B. Y.; Ravichander, A.; Lu, X.; Dziri, N.; Sclar, M.; Chandu, K.; Bhagavatula, C.; and Choi, Y. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*.

Liu, W.; Qiu, Z.; Feng, Y.; Xiu, Y.; Xue, Y.; Yu, L.; Feng, H.; Liu, Z.; Heo, J.; Peng, S.; et al. 2023a. Parameter-efficient orthogonal finetuning via butterfly factorization. *arXiv preprint arXiv:2311.06243*.

Liu, W.; Wang, X.; Wu, M.; Li, T.; Lv, C.; Ling, Z.; Zhu, J.; Zhang, C.; Zheng, X.; and Huang, X. 2023b. Aligning large language models with human preferences through representation engineering. *arXiv preprint arXiv:2312.15997*.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 746–751.

Nanda, N.; Lee, A.; and Wattenberg, M. 2023. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*.

Park, K.; Choe, Y. J.; and Veitch, V. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Qi, X.; Qiu, J.; Juan, X.; Wu, Y.; and Wang, M. 2025. Shallow Preference Signals: Large Language Model Aligns Even Better with Truncated Data? *arXiv preprint arXiv:2505.17122*.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *Language Modeling*.

Singh, S.; Ravfogel, S.; Herzig, J.; Aharoni, R.; Cotterell, R.; and Kumaraguru, P. 2024. Mimic: Minimally modified counterfactuals in the representation space. *Computer Research Repository*.

Tang, X.; Wang, X.; Lv, Z.; Min, Y.; Zhao, W. X.; Hu, B.; Liu, Z.; and Zhang, Z. 2025. Unlocking General Long Chain-of-Thought Reasoning Capabilities of Large Language Models via Representation Engineering. *arXiv preprint arXiv:2503.11314*.

Tohma, K.; Okur, H. İ.; Gürsoy-Demir, H.; Aydın, M. N.; and Yeroğlu, C. 2025. SmartControl: Interactive PID controller design powered by LLM agents and control system expertise. *SoftwareX*.

Wang, C.; Li, J.; Chen, Y.; Liu, K.; and Zhao, J. 2025. A survey of recent advances in commonsense knowledge acquisition: Methods and resources. *Machine Intelligence Research*, 22(2): 201–218.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 24824–24837.

Wu, M.; Liu, W.; Wang, X.; Li, T.; Lv, C.; Ling, Z.; Zhu, J.; Zhang, C.; Zheng, X.; and Huang, X. 2024a. Advancing parameter efficiency in fine-tuning via representation editing. *arXiv preprint arXiv:2402.15179*.

Wu, Z.; Arora, A.; Wang, Z.; Geiger, A.; Jurafsky, D.; Manning, C. D.; and Potts, C. 2024b. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Xu, N.; Xue, D.; Qian, S.; Fang, Q.; and Hu, J. 2025. Prompting large language models for automatic question tagging. *Machine Intelligence Research*, 1–12.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhang, F.; Li, L.; Chen, J.; Jiang, Z.; Wang, B.; and Qian, Y. 2023. Increlora: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv preprint arXiv:2308.12043*.

Zhang, R.; Han, J.; Liu, C.; Zhou, A.; Lu, P.; Qiao, Y.; Li, H.; and Gao, P. 2024. LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *The Twelfth International Conference on Learning Representations*.

Zhang, Y.; Tang, X.; and Hao, K. 2025. CRMR: A Collaborative Multistep Reasoning Framework for Solving Mathematical Problems. *Machine Intelligence Research*, 1–14.

Zhu, F.; Dai, D.; and Sui, Z. 2025. Language models encode the value of numbers linearly. In *International Conference on Computational Linguistics*, 693–709.

Zou, A.; Phan, L.; Chen, S. L.; Campbell, J.; Guo, P. H.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.; et al. 2024. Enhancing Neural Network Transparency through Representation Analysis. *arXiv preprint*.