PT-MoE: An Efficient Finetuning Framework for Integrating Mixture-of-Experts into Prompt Tuning

Zongqian Li [1] **Yixuan Su** [1] **Nigel Collier** [1]

Abstract

Parameter-efficient fine-tuning (PEFT) methods have shown promise in adapting large language models, yet existing approaches exhibit counter-intuitive phenomena: integrating router into prompt tuning (PT) increases training efficiency yet does not improve performance universally; parameter reduction through matrix decomposition can improve performance in specific domains. Motivated by these observations and the modular nature of PT, we propose PT-MoE, a novel framework that integrates matrix decomposition with mixture-of-experts (MoE) routing for efficient PT. Results across 17 datasets demonstrate that PT-MoE achieves state-of-the-art performance in both question answering (QA) and mathematical problem solving tasks, improving F1 score by 1.49 points over PT and 2.13 points over LoRA in QA tasks, while enhancing mathematical accuracy by 10.75 points over PT and 0.44 points over LoRA, all while using 25% fewer parameters than LoRA. Our analysis reveals that while PT methods generally excel in QA tasks and LoRA-based methods in math datasets, the integration of matrix decomposition and MoE in PT-MoE yields complementary benefits: decomposition enables efficient parameter sharing across experts while MoE provides dynamic adaptation, collectively enabling PT-MoE to demonstrate cross-task consistency and generalization abilities. These findings, along with ablation studies on routing mechanisms and architectural components, provide insights for future PEFT methods. [1]

1. Introduction

Large language models (LLMs) have shown remarkable capabilities but require improvements in efficiency across data (Li & Cole, 2025), training, and inference (Li et al., 2024; 2025a;b). PEFT methods address training efficiency challenge by updating only a small subset of parameters (Han et al., 2024). **Prompt tuning** (PT) stands out among PEFT approaches with its unique advantages: minimizing trainable parameters through soft prompt optimization, enabling modular utilization through task-specific prompts without model modifications, and supporting flexible knowledge composition (Lester et al., 2021). These properties make it effective for low-resource and multi-task applications where efficient adaptation is essential.

Despite these advantages, we observe two **counter-intuitive phenomena** in prompt tuning. First, integrating router into prompt tuning does not decrease training efficiency yet improves performance in specific domains rather than universally (SMoP vs PT in Table 2), suggesting domain-dependent optimization dynamics. **Second**, decomposing soft prompts into low-rank matrices, while reducing parameters, can surprisingly improve model performance in specific areas (DPT vs PT in Table 4). These phenomena indicate that the relationship between parameter efficiency and model effectiveness in prompt tuning is more nuanced than previously understood, motivating the need for a more sophisticated approach to prompt optimization.

Based on these observations, we propose a novel framework, **Prompt Tuning with Efficient Mixture-of-Experts (PT-MoE)**, that combines matrix decomposition with MoE routing. As shown in Figure 1, our approach not only achieves state-of-the-art performance, but also uses minimal trainable parameters and moderate training steps.

Our work makes three key **contributions**:

- **Novel finetuning framework:** We propose PT-MoE, integrating matrix decomposition with MoE for prompt tuning. Our framework achieves state-of-the-art performance with fewer parameters while outperforming either method alone, demonstrating their complementary benefits.
- **Design dynamics:** We thoroughly analyze key variables

[1]University of Cambridge. Correspondence to: Zongqian Li, Nigel Collier <zl510@cam.ac.uk, nhc30@cam.ac.uk>.

Workshop of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada.

[1]https://github.com/ZongqianLi/PT-MoE

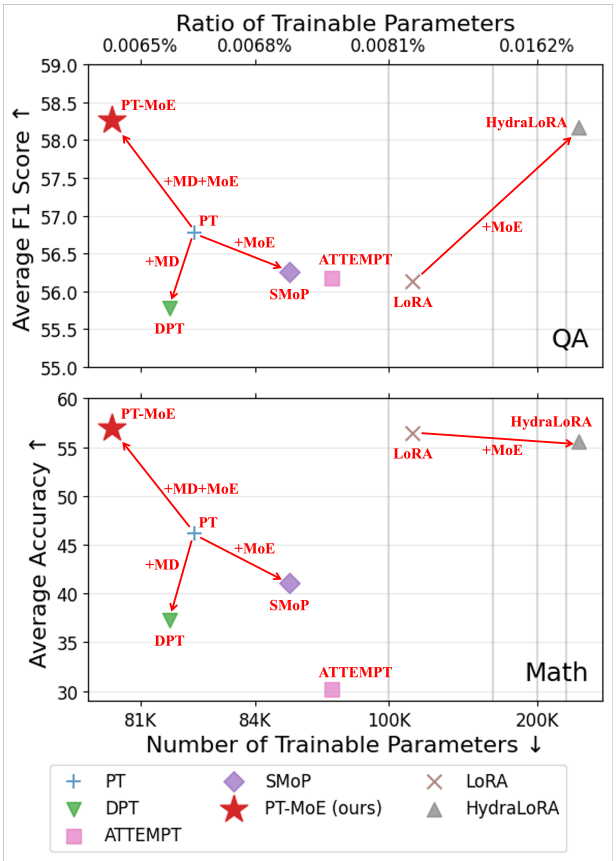

Figure 1: **Performance and parameter efficiency comparison** of PEFT methods on QA and mathematical tasks. The upper subgraph shows average F1 scores on 12 MRQA benchmark datasets, while the lower subgraph shows average accuracy on 5 mathematical datasets. The x-axis is the number of trainable parameters, with corresponding parameter ratio shown at the top. ↑ indicates higher is better; ↓ indicates lower is better. Red arrows indicate method transformations: +MD (matrix decomposition), +MoE (mixture-of-experts), or their combination. PT excels in QA tasks while LoRA demonstrates advantages in mathematical tasks. PT-MoE achieves the best performance on both task types while using fewer parameters than alternative methods, demonstrating that combining matrix decomposition and MoE yields complementary benefits despite each component individually decreasing performance when applied to PT.

influencing the performance of PT-MoE, including prompt length, expert count, trainable parameters, routing mechanisms, and model size. Findings provide design guidelines for future parameter-efficient tuning approaches.

- **Key insights:** Our comprehensive analysis across diverse tasks reveals several important findings: **First**, prompt tuning methods excel in QA tasks while LoRA-based methods demonstrate advantages in mathematical reasoning; **Second**, matrix decomposition reduces parameters while potentially improving domain-specific performance, whereas MoE integration increases parameter count without compromising training efficiency; and **Third**, combining matrix decomposition and MoE enables PT-MoE to achieve superior performance across all tasks while maintaining minimal parameter count and moderate training costs, whereas applying either of them individually can decrease average performance.

The remainder of this paper is organized as follows: Section 2 reviews related work in prompt tuning, covering both direct tuning approaches and transfer learning methods. Section 3 presents our PT-MoE framework, detailing the matrix decomposition strategy, dynamic router design, and training methodology. Section 4 describes our experimental design across QA and mathematical problem-solving tasks. Section 5 presents comprehensive results, including detailed ablation studies analyzing the influence of prompt length, parameter count, expert number, routing mechanisms, and model size, followed by efficiency analysis. Section 6 concludes with key findings and future directions.

2. Related Work

To contextualize our approach, we review existing prompt tuning methods, which fall into two categories: direct prompt tuning approaches focusing on architectural innovations, and transfer learning methods enabling cross-task knowledge sharing.

Direct prompt tuning methods have developed into four main branches: (1) General approaches that directly optimize prompt parameters, including Prompt Tuning that prepends trainable vectors to input while freezing the language model (Lester et al., 2021), XPrompt that employs hierarchical structured pruning to identify and retain important prompt tokens (Ma et al., 2022), and P-Tuning v2 that introduces deep prompts across all transformer layers (Liu et al., 2022); (2) Encoder-based methods that leverage additional modules, such as P-Tuning that incorporates an encoder to learn dependencies between continuous embeddings (Liu et al., 2023), Residual Prompt Tuning (RPT) that employs a residual part with down/up-projection layers for stable optimization (Razdaibiedina et al., 2023), and Prefix Tuning that prepends trainable key-value pairs at each layer through a reparameterization section (Li & Liang, 2021); (3) Decomposition methods that decompose prompt embeddings, including Decomposed Prompt Tuning (DPT) that applies low-rank matrix decomposition to reduce parameter count (Xiao et al., 2023), and DePT that combines shorter soft prompts with low-rank updates to word embeddings (Shi & Lipani, 2024); and (4) MoE approaches such as Sparse Mixture-of-Prompts (SMoP) that employs multiple

shorter prompts with a dynamic router to route inputs to the most suitable soft prompt (Choi et al., 2023).

Transfer learning approaches in prompt tuning have developed into three categories: (1) General approaches that directly transfer prompt knowledge, including SPoT that introduces both generic transfer through multi-task pre-training and targeted transfer via task similarity matching (Vu et al., 2022), and ATTEMPT that dynamically combines multiple source prompts through an attention-based mixing mechanism with instance-level adaptation (Asai et al., 2022); (2) Encoder-based methods that facilitate knowledge transfer through additional modules, such as TransPrompt that employs parallel task-specific and universal encoders with balancing mechanisms for obtaining both task-dependent and task-agnostic knowledge (Wang et al., 2021), and Cross-Task Prompt Tuning (CTPT) that leverages multi-head attention for cross-task knowledge transfer with dimension reduction and derivative-free optimization (Xu et al., 2023); and (3) Decomposition methods exemplified by Multitask Prompt Tuning (MPT) that decomposes prompts into shared and task-specific components through knowledge distillation, enabling efficient transfer while preserving task-specific adaptability through a rank-one decomposition strategy (Wang et al., 2023).

3. Methods

Building upon the insights from prior work, we propose a new parameter-efficient prompt tuning framework, PT-MoE, shown in Figure 2 and Algorithm 1.

Framework Overview. PT-MoE integrates matrix decomposition and dynamic routing. Given an input sequence $\mathbf{x}$, our framework first generates routing weights $\mathbf{w}$ through a router R: $\mathbf{w} = R(\mathbf{x})$. These weights determine the selection among N decomposed prompts, where each prompt $\mathbf{P}_i$ is decomposed as $\mathbf{P}_i = \mathbf{A}_i\mathbf{B}$, with $\mathbf{A}_i$ being prompt-specific and $\mathbf{B}$ being shared across all prompts. The final soft prompt $\mathbf{P}$ is computed as $\mathbf{P} = \sum_{i=1}^{N} w_i\mathbf{A}_i\mathbf{B}$, which is then prepended to the input sequence for the frozen language model.

Matrix Decomposition. To achieve parameter efficiency, we decompose each prompt matrix $\mathbf{P}_i \in \mathbb{R}^{T\times H}$ into a prompt-specific matrix $\mathbf{A}_i \in \mathbb{R}^{T\times R}$ and a shared matrix $\mathbf{B} \in \mathbb{R}^{R\times H}$, where T, H, and R denote the prompt length, hidden dimension, and low-rank dimension respectively. This reduces parameters from $O(NTH)$ to $O(NTR+RH)$ for N prompts. The low-rank dimension R is either manually determined or computed to maintain parameter efficiency. For initialization, we first transform task-relevant text into word embeddings $\mathbf{E} \in \mathbb{R}^{T\times H}$, then perform SVD: $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Each $\mathbf{A}_i$ is initialized as $\mathbf{U}$: $R\mathbf{\Sigma}R^{1/2}$ and the shared $\mathbf{B}$ as $\mathbf{\Sigma}R^{1/2}\mathbf{V}^\top_{R:}$, where subscript R indicates truncation to the first R components. This approach ensures the initial prompts have task-relevant information while maintaining the parameter efficiency of decomposition.

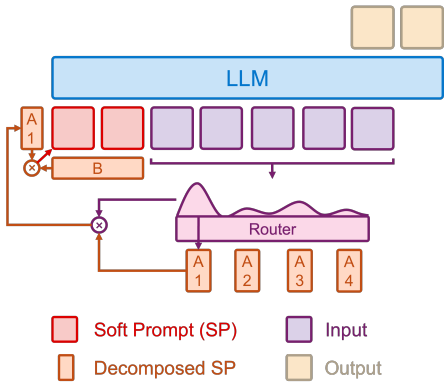

Figure 2: **Framework** of PT-MoE. Each soft prompt is decomposed into an input-specific matrix A_i and a shared matrix B, with a router adaptively selecting and combining prompt components based on input. The resulting soft prompt is prepended to the input for the frozen LLM.

Algorithm 1 Pseudo code of PT-MoE

Require: Base model $\mathcal{M}$; input batch $X = x_1, ..., x_b$; parameters θ
Notation: b - batch size; s - sequence length; n - number of prompts; k - tokens per prompt; d - low-rank dimension; h - hidden dimension
for batch $x \in X$ **do**
  Get input embeddings $E = \mathcal{M}_{\text{embed}}(x)$ *where* $E \in \mathbb{R}^{b\times s\times h}$
  Calculate mean embeddings $\mu = \text{mean}(E, \text{dim} = 1)$ *where* $\mu \in \mathbb{R}^{b\times h}$
  Compute router logits $l = W\mu + b$ *where* $W \in \mathbb{R}^{n\times h}, b \in \mathbb{R}^n, l \in \mathbb{R}^{b\times n}$
  Get router weights $w = \text{softmax}(l)$ *where* $w \in \mathbb{R}^{b\times n}$
  for each sample j in batch **do**
    Find indices of top-k weights: $i_{topk} = \text{argsort}(w_j)[-k:]$
    Zero all weights except top-k: $w_j[i] = 0$ for all $i \notin i_{topk}$
  end for
  Initialize prompt embeddings $P = 0, P \in \mathbb{R}^{b\times k\times d}$
  for each weight w_i in w **do**
    Compute weighted prompts $P = P + w_iA_i$ *where* $A_i \in \mathbb{R}^{k\times d}$
  end for
  Project to model dimension $P = P \times B$ *where* $B \in \mathbb{R}^{d\times h}$
  Combine with input: $C = \text{concat}(P, E)$ *where* $C \in \mathbb{R}^{b\times(k+s)\times h}$
  Generate through base model: $y = \mathcal{M}(C)$
end for
Ensure: Model predictions y

MRQA (Extractive QA)	
In-domain	SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), NaturalQuestions (Kwiatkowski et al., 2019)
Out-of-domain	BioASQ (Partalas et al., 2013), DROP (Dua et al., 2019), DuoRC (Saha et al., 2018), RACE (Lai et al., 2017), RelationExtraction (Levy et al., 2017), TextbookQA (Kembhavi et al., 2017)
Mathematics (Problem Solving)	
In-domain	GSM8K (Cobbe et al., 2021)
Out-of-domain	SVAMP: Subtraction, Addition, Common-Division, Multiplication (Patel et al., 2021); ASDIV (Miao et al., 2020); MAWPS (Koncel-Kedziorski et al., 2016); MATH_PROBLEMS (Nebrelbug, 2024)

Table 1: Overview of training and evaluation **datasets** that span two task categories: extractive QA (MRQA benchmark with 12 QA datasets) and mathematical problem solving (GSM8K and specific mathematical datasets). For each category, datasets are divided into in-domain sets used for training, validation, and evaluation, and out-of-domain sets used exclusively for testing generalization ability.

Dynamic Router. The router adaptively selects prompts based on input context. Given an input sequence embedding $\mathbf{x} \in \mathbb{R}^H$ (obtained by averaging token embeddings), the router computes logits through a linear projection: $\mathbf{l} = \mathbf{W}\mathbf{x} + \mathbf{b}$, where $\mathbf{W} \in \mathbb{R}^{N \times H}$ and $\mathbf{b} \in \mathbb{R}^N$. During training, we apply multiplicative Gaussian noise to encourage exploration: $\mathbf{l}' = \mathbf{l} \odot (1 + \epsilon)$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The routing weights are computed as $\mathbf{w} = \text{softmax}(\mathbf{l}') \odot \mathbf{1}_{\text{argmax}}$, where $\mathbf{1}_{\text{argmax}}$ is a one-hot vector with 1 at the position of the maximum value. This hard selection strategy reduces overlap between prompts while maintaining end-to-end differentiability through straight-through estimation.

Training and Prediction. During training, we optimize both the router parameters and decomposed prompt matrices while keeping the base model frozen. For language model training, we use negative log-likelihood loss computed only on non-prompt positions using a binary mask: $\mathcal{L} = -\sum_{t \in \mathcal{M}} \log p(y_t | x_{<t})$, where $\mathcal{M}$ denotes non-prompt positions. We employ AdamW optimizer with warmup followed by a constant learning rate schedule, and gradient accumulation for stable optimization. At inference, noise is not added in the router, ensuring deterministic prompt selection.

4. Experimental Design

4.1. Datasets

We complete evaluations across **17** diverse datasets, as shown in Table 1, where in-domain datasets are split into training, validation, and test sets, while out-of-domain datasets are used exclusively for testing. For **QA**, we utilize 12 MRQA datasets (Fisch et al., 2019), with in-domain sets like SQuAD (Rajpurkar et al., 2016) testing information extraction abilities and out-of-domain sets like DROP (Dua et al., 2019) evaluating domain adaptation. For mathematical **problem solving**, we use GSM8K (Cobbe et al., 2021) from MetaMath (Yu et al., 2024) as our in-domain dataset, complemented by specific out-of-domain datasets including all the subsets of SVAMP (Patel et al., 2021), ASDIV (Miao et al., 2020), MAWPS (Koncel-Kedziorski et al., 2016), and MATHPROBLEMS (Nebrelbug, 2024).

4.2. Gold Standard and Baselines

We employ full model fine-tuning as our **gold standard**, which updates all parameters but requires substantial computational resources. Our **baselines**[2] include typical methods from **prompt tuning categories**: For **direct prompt tuning**, we select (1) PT from general approaches, (2) DPT from decomposition methods, and (3) SMoP from MoE approaches. While **transfer learning** methods like (4) ATTEMPT typically involve multi-turn training, we also evaluate its architecture under similar training for comprehensive comparison. We additionally compare **other PEFT methods** including (5) LoRA and (6) HydraLoRA, with HydraLoRA adopting a MoE-like architecture that uses a shared down-projection matrix and multiple routed up-projection matrices. **These two LoRA-based methods require model architecture modifications unlike the modular nature of prompt tuning methods.**

4.3. Evaluation Metrices

We employ task-specific evaluation metrics. For extractive QA tasks from MRQA, we adopt two metrics: **F1** score, which evaluates the token-level overlap between predicted and ground truth answer spans, balancing precision and recall; and **Exact Match (EM)**, which measures the percentage of predictions that exactly match the ground truth. For mathematical problem solving tasks, we use **accuracy**, defined as the percentage of correctly solved problems with exact answer matches.

[2]All methods are controlled to have similar parameter budgets, with detailed configurations shown in Table 9 of the Appendix.

Method	# para.	In-domain						Out-of-domain						Avg.
		SQ	News	Tri	Srch	HP	NQ	BSQ	DP	DRC	RC	RE	TB	
FT	1.2B	78.76	48.69	71.04	71.35	72.96	67.56	70.19	43.87	48.11	43.44	81.60	52.71	62.52
LoRA	106k	69.82	39.91	70.61	55.56	63.29	65.92	65.38	35.25	43.69	38.04	74.09	52.00	56.13
HydraLoRA	278k	74.24	44.05	71.38	60.13	64.02	66.31	68.76	34.38	44.36	40.00	77.97	52.44	58.17
PT	81k	72.31	48.18	65.93	49.74	58.69	62.18	68.59	40.39	43.30	42.10	82.43	47.34	56.77
DPT	81k	70.99	48.42	65.41	46.94	58.49	61.65	65.56	38.80	43.64	41.89	80.85	46.62	55.77
SMoP	86k	74.15	**48.96**	66.13	41.08	58.96	61.17	68.59	39.92	42.07	42.34	83.73	**47.85**	56.25
ATTEMPT	90k	**74.22**	48.18	65.31	37.64	60.18	59.59	66.69	45.32	42.86	**43.01**	**84.11**	46.91	56.17
PT-MoE	**80k**	73.85	48.24	**67.34**	**51.33**	**62.16**	**62.95**	**69.33**	**48.02**	**43.96**	42.51	83.70	45.71	**58.26**

Table 2: Evaluation results (F1 scores) for various PEFT methods on **QA** datasets. SQ: SQuAD; News: NewsQA; Tri: TriviaQA; Srch: SearchQA; HP: HotpotQA; NQ: NaturalQuestions; BSQ: BioASQ; DP: DROP; DRC: DuoRC; RC: RACE; RE: RelationExtraction; TB: TextbookQA. The bold values indicate the best performance among prompt tuning-based methods.

Method	# para.	In-domain						Out-of-domain						Avg.
		SQ	News	Tri	Srch	HP	NQ	BSQ	DP	DRC	RC	RE	TB	
FT	1.2B	65.28	32.76	62.29	61.50	56.19	50.45	49.06	32.26	38.84	29.52	66.99	43.71	49.07
LoRA	106k	56.26	25.26	64.11	46.10	47.48	49.54	42.02	25.48	33.24	24.92	58.58	44.17	43.09
HydraLoRA	278k	61.63	27.80	64.32	50.06	47.73	49.59	44.01	24.75	33.57	26.11	62.68	43.97	44.69
PT	81k	61.25	32.62	59.49	42.40	44.45	47.28	51.79	30.60	34.64	29.82	72.45	39.52	45.52
DPT	81k	58.49	**32.88**	58.56	39.65	44.33	46.54	49.46	28.74	35.64	30.26	70.48	38.72	44.48
SMoP	86k	63.15	32.81	59.48	34.51	43.80	46.39	50.06	29.94	34.11	30.56	74.59	**40.25**	44.97
ATTEMPT	90k	**63.71**	32.50	58.71	31.24	45.77	45.66	49.26	36.06	34.84	30.41	**75.13**	39.52	45.23
PT-MoE	**80k**	63.34	32.85	**60.87**	**43.98**	**47.29**	**48.18**	**52.06**	**37.12**	**35.64**	**31.75**	74.18	38.25	**47.13**

Table 3: Evaluation results (Exact Match) for **QA** datasets.

4.4. Models

We get our main results using **LLaMA-3.2-1B-Instruct** as the base model for fine-tuning methods (Grattafiori et al., 2024). For ablation studies on model size, we additionally employ **LLaMA-3.2-3B-Instruct**.

5. Results

5.1. Question Answering

The results on MRQA datasets shown in Table 2 and 3 demonstrate the effectiveness of PT-MoE across various QA tasks. We highlight seven key findings: (1) PT-MoE achieves superior overall performance with an average F1 score of 58.26%, outperforming SMoP (56.25%) by 2.01 points and the standard PT (56.77%) by 1.49 points, establishing a new state-of-the-art on the MRQA benchmark. (2) This improvement is further validated by Exact Match metrics, where PT-MoE demonstrates even more gains (47.13% for average, outperforming SMoP and PT by 2.16 and 1.61 points respectively). (3) PT-MoE exhibits strong generalization abilities across both in-domain and out-of-domain scenarios. It achieves the highest performance on four out of six in-domain datasets and three out of six out-of-domain datasets. (4) The stability of PT-MoE is evidenced by consistent improvements over PT across 11 out of 12 datasets, with only marginal decreases in the RACE dataset. In contrast, SMoP shows performance decrease on 5 datasets compared to PT. (5) Individual architectural components show limited gains: both matrix decomposition (DPT, 55.77% F1) and MoE (SMoP, 56.25% F1) underperform standard prompt tuning (PT, 56.77% F1). (6) PT-MoE's integration of matrix decomposition and MoE yields complementary benefits, outperforming both DPT and SMoP by 2.49 and 2.01 points for F1 respectively. This improvement over individual approaches proves the mutually beneficial nature of these methods. (7) Notably, while PT-MoE achieves lower overall performance than FT, it reaches comparable or even higher scores than FT on specific datasets such as DROP (48.02% vs 43.87% F1) while using only 80K parameters compared to FT's 1.2B. These results collectively validate the effectiveness of the architectural design of PT-MoE and demonstrate its superior performance in accuracy and generalization across diverse QA scenarios.

5.2. Mathematical Problem Solving

The results on mathematical tasks reveal several characteristics compared to QA tasks. We highlight six key findings: (1) PT-MoE achieves state-of-the-art performance with an average accuracy of 56.91%, improving upon PT (46.16%) by 10.75 points, demonstrating its effectiveness in mathematical reasoning. (2) The benefits of MoE integration shows method-dependent characteristics: in prompt tuning approaches, PT-MoE and SMoP show different changes over PT (by +10.75 and -5.11 points respectively); when applied to LoRA methods, HydraLoRA shows slightly performance decrease compared to LoRA. (3) LoRA-based methods demonstrate advantages in mathematical tasks compared to their performance in QA. While LoRA underperformed PT by 5.36 points in MRQA, it outperforms PT by 10.31 points in mathematical tasks, indicating task-specific

Method	# para.	In-domain	Out-of-domain								Average
		GSM8K	Subtraction	Addition	Division	Multiplication	SVAMP	ASDIV	MAWPS	MP500	
FT	1.2B	58.15	68.75	64.40	62.50	48.48	61.03	86.04	82.53	30.60	63.67
LoRA	106k	41.77	67.50	61.01	52.08	33.33	53.48	73.42	70.70	43.00	56.47
HydraLoRA	278k	41.31	57.50	62.71	52.08	39.39	52.92	74.08	76.05	33.40	55.55
PT	81k	34.11	41.87	50.84	66.66	33.33	48.18	60.13	57.18	31.20	46.16
Decomp. PT	81k	26.08	43.12	35.59	64.58	27.27	42.64	56.14	43.09	18.20	37.23
SMoP	86k	27.97	38.12	35.59	33.33	33.33	35.09	49.50	65.91	26.80	41.05
ATTEMPT	90k	27.36	40.00	35.59	37.50	27.27	35.09	24.91	49.01	14.60	30.19
PT-MoE	**80k**	**35.63**	**55.62**	**55.93**	**79.16**	**36.36**	**56.77**	**77.74**	**71.83**	**42.60**	**56.91**

Table 4: Accuracy (%) on **mathematical problem-solving** tasks with the number of trainable parameters shown in the second column. The first four out-of-domain datasets are from the SVAMP dataset. MP500 denotes the first 500 questions from MATH_PROBLEMS.

strengths of different PEFT approaches. (4) PT-MoE demonstrates unique cross-task consistency: while prompt tuning methods excel in QA tasks and LoRA-based methods in mathematical tasks, PT-MoE achieves the highest average performance in both domains, indicating robust adaptability across different problem types. (5) While PEFT methods consistently underperform full fine-tuning, the performance gap is larger in mathematical tasks compared to QA tasks, with a wider performance range among different methods. Notably, PT-MoE achieves comparable or higher performance to full fine-tuning on specific datasets such as Division and MP500. (6) PT-MoE demonstrates superior parameter efficiency, achieving higher performance than LoRA while using only 75% of its parameters (80k vs 106k), and outperforming HydraLoRA which uses 3.5 times more parameters. These findings highlight both the unique challenges of mathematical tasks and the robust adaptability of PT-MoE across different problem domains.

5.3. Case Study

To better understand the performance characteristics of PT-MoE, we present a detailed case study of polynomial addition in Table 5. In this example, the response of the base model exhibits information loss, specifically omitting the linear term during simplification steps, leading to an incorrect final result. The conventional prompt tuning approach exhibits hallucinations and conceptual errors, particularly in degree identification and term combination, resulting in wrong terms like $2y^4$ and $-6y^3$. PT-MoE maintains information completeness throughout the solution process and avoids hallucinations, ultimately producing the correct polynomial expression. Notably, PT-MoE achieves this with a more concise solution process, demonstrating efficient problem-solving steps while maintaining accuracy.

5.4. Ablation Studies

To comprehensively evaluate the design choices in PT-MoE, we conduct ablation studies on **five influencing variables**: soft prompt length, trainable parameters, number of experts, routing mechanisms, and model size. For each variable, we keep other variables fixed at their default values (soft prompt length=40, trainable parameters≈80K, number of experts=2, probationary-selective routing, 1B base model) while varying the target component to identify its influence on model performance.

Soft prompt length. We evaluate prompt lengths ranging from 20 to 80 tokens (Figure 3 Left). Three consistent observations appear: (1) In-domain performance exceeds out-of-domain across all lengths, maintaining a 5-6% F1 score margin; (2) Both domains achieve optimal performance at 40 tokens, with peak F1 scores of 60.66% and 55.28% respectively; and (3) Performance in both domains follows a similar trend, improving up to 40 tokens then decreasing. These findings indicate that the optimal prompt length is domain-agnostic, though the absolute performance levels remain domain-dependent.

Number of experts. We investigate the influence of expert count by varying it from 1 to 8 (Figure 3 Center left). There are three key points: (1) Single-expert configuration yields the poorest performance (58.90% and 52.64% F1 for in-domain and out-of-domain), demonstrating the necessity of MoE; (2) Performance exhibits an initial increase followed by decrease, with in-domain peaking at N=2 (60.66% F1) and out-of-domain at N=4 (55.84% F1), suggesting different optimal routing abilities for each domain; (3) In-domain tasks consistently outperform out-of-domain scenarios by a 4-6% F1 margin across all expert counts. These observations demonstrate that the optimal number of experts varies by domain type and highlight the importance of balancing expert focus with routing difficulty.

Trainable parameters. We vary the parameter count from 18K to 163K to analyze its influence on model performance (Figure 3 Center Right). Three key observations appear: (1) Performance consistently improves with increasing parameters, from 57.51% to 61.04% F1 for in-domain and 53.96% to 55.38% F1 for out-of-domain tasks, and notably maintains stability without decrease even at higher parameter counts, differing from conventional prompt tuning methods; (2) While both in-domain and out-of-domain tasks show increasing trend, they exhibit different parameter de-

Question: Let $f(y) = y^4 - 3y^3 + y - 3$ and $g(y) = y^3 + 7y^2 - 2$. Find $f(y) + g(y)$. Write your answer as a polynomial with terms of decreasing degree. **Standard Answer:** $y^4 - 2y^3 + 7y^2 + y - 5$

Original:	**Prompt Tuning:**	**PT-MoE:**
## Step 1: Identify the polynomials f(y) and g(y) The polynomial f(y) is given as $f(y) = y^4 - 3y^3 + y - 3$ and the polynomial g(y) is given as $g(y) = y^3 + 7y^2 - 2$. ## Step 2: Add the polynomials f(y) and g(y) To find the sum of f(y) and g(y), we need to add the corresponding terms of each polynomial. ## Step 3: Perform the addition $f(y) + g(y) = (y^4 - 3y^3 + y - 3) + (y^3 + 7y^2 - 2)$ ## Step 4: Combine like terms Combine the terms with the same degree to simplify the expression. ## Step 5: Simplify the expression $f(y)+g(y) = y^4-3y^3+y^3+7y^2-2-3$ ## Step 6: Combine like terms Combine the terms with the same degree. ## Step 7: Simplify the expression $f(y) + g(y) = y^4 - 2y^3 + 7y^2-2$ ## Step 8: Write the final answer The final answer is: $y^4 - 2y^3 + 7y^2-2$	## Step 1: To find the sum of the two polynomials $f(y)$ and $g(y)$, we need to add their corresponding terms. ## Step 2: The polynomial $f(y) = y^4 - 3y^3 + y - 3$ has terms of degree 4, 3, 2, and 1, while the polynomial $g(y) = y^3 + 7y^2 - 2$ has terms of degree 3, 2, and 1. ## Step 3: When we add these two polynomials, we need to combine like terms, which means adding the coefficients of terms with the same degree. ## Step 4: The terms with degree 4 are y^4 and y^4, so their sum is $2y^4$. ## Step 5: The terms with degree 3 are $-3y^3$ and $-3y^3$, so their sum is $-6y^3$. ## Step 6: The terms with degree 2 are y and $7y^2$, so their sum is $7y^2$. ## Step 7: The terms with degree 1 are 1 and -2, so their sum is -1. ## Step 8: Therefore, the sum of the two polynomials $f(y)$ and $g(y)$ is $2y^4 - 6y^3 + 7y^2-1$. ## Step 9: The answer is $2y^4 - 6y^3 + 7y^2-1$.	## Step 1: Identify the polynomials f(y) and g(y) The given polynomials are $f(y) = y^4 - 3y^3 + y - 3$ and $g(y) = y^3 + 7y^2 - 2$. ## Step 2: Add the polynomials f(y) and g(y) To find $f(y) + g(y)$, we need to add the corresponding terms of the two polynomials. ## Step 3: Perform the addition $f(y) + g(y) = (y^4 - 3y^3 + y - 3) + (y^3 + 7y^2 - 2)$ ## Step 4: Combine like terms Combine the terms with the same degree of y: $y^4 + (y^3 - 3y^3) + (y^3 + 7y^2) + (y - 3) - 2$ ## Step 5: Simplify the expression Simplify the expression by combining the coefficients of like terms: $y^4 - 2y^3 + 7y^2 + y - 5$ ## Step 6: Write the final expression The expression $f(y)+g(y)$ is $y^4-2y^3+7y^2 + y - 5$. ## Note: y is the variable, and the answer is $y^4 - 2y^3 + 7y^2 + y - 5$.

Table 5: A **case study** comparing three solution approaches (Original, Prompt Tuning, and PT-MoE) for a polynomial addition problem. Errors in the outputs are highlighted in red (incorrect terms), orange (missing terms), and blue (hallucinated terms).

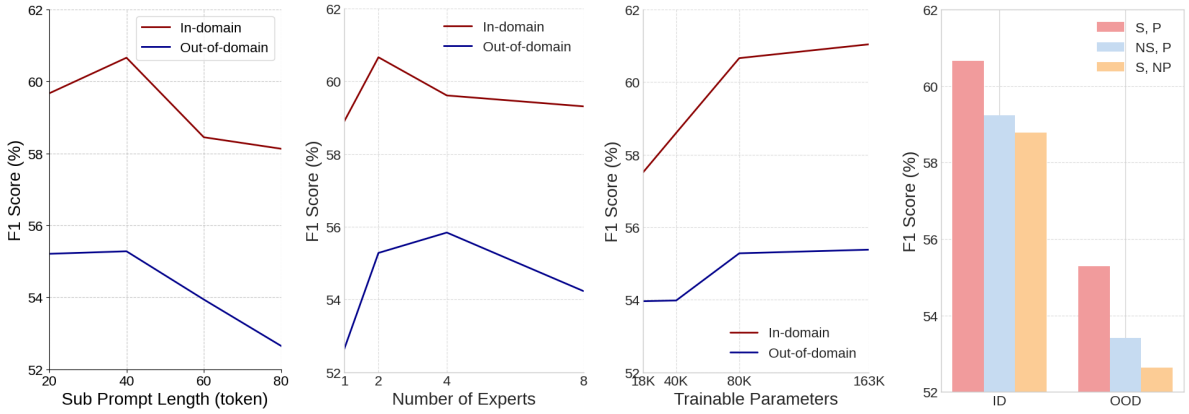

Figure 3: **Ablation studies** on key components of PT-MoE, showing the influence of (Left) prompt length, (Center left) number of experts, (Center right) trainable parameters, and (Right) routing mechanisms ((N)S: (Non-)Selective, (N)P: (Non-)Probationary) on in-domain (ID) and out-of-domain (OOD) performance.

pendence behaviours, in-domain tasks demonstrate rapid improvement before 80K parameters, while out-of-domain tasks show accelerated growth in the 40K-80K range; (3) In-domain performance maintains a consistent advantage over out-of-domain tasks across all parameter ranges, with F1 scores differing by approximately 4-6%. These findings show that PT-MoE effectively leverages additional parameters to achieve continuous performance gains.

Routing mechanisms. We examine two key routing design choices (Figure 3 Right): selective routing, which uses only the highest-weighted expert versus non-selective routing that utilizes all experts with their respective weights, and probationary routing, which multiplies the output by the

	PT	SMoP	PT-MoE
GSM8K	56.70	**61.78**	59.74
SVAMP	69.36	**74.69**	72.81
ASDIV	76.41	80.06	**81.39**
MAWPS	70.70	70.70	**78.02**
MP500	59.00	60.80	**63.60**
Average	66.43	69.61	**71.11**

Table 6: Performance comparison (accuracy %) of standard and MoE-based prompt tuning methods on mathematical problem solving tasks using a **3B** base model.

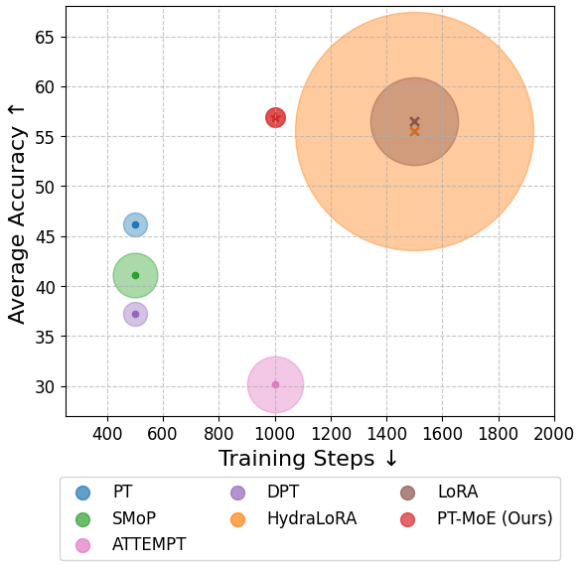

Figure 4: Parameter and training **efficiency comparison** across different methods. The x-axis shows training steps for the highest performance after training parameter search, while the y-axis shows the average accuracy on math datasets. Circle sizes indicate the number of trainable parameters, with larger circles indicating more parameters.

router's selection probability versus non-probationary routing that uses original outputs. Our results show four key findings: (1) The combination of selective and probationary routing (S, P) consistently outperforms other configurations (NS, P and S, NP) across both in-domain (60.66% vs 59.24% and 58.78% F1) and out-of-domain tasks (55.28% vs 53.41% and 52.64% F1), suggesting the complementary benefits of focused expert utilization and confidence-based output; (2) Probationary routing demonstrates superior performance over its non-probationary counterpart, indicating the value of incorporating router confidence in the final output; (3) Under probationary conditions, selective routing achieves 1.42% higher F1 score while reducing utilized parameters compared to non-selective routing, highlighting the effectiveness and efficiency of domain-specific knowledge; (4) All routing configurations maintain higher performance on in-domain tasks compared to out-of-domain scenarios, though the relative performance rankings remain consistent across domains. These findings collectively demonstrate that the selective probationary routing mechanism achieves an optimal balance between model performance and computational efficiency.

Model size. We conduct additional studies using a 3B version of the base model, comparing PT-MoE with PT and the MoE-integrated method, SMoP (Table 6). Three key findings are found: (1) PT-MoE maintains its advantage at larger sizes, achieving the highest average accuracy of 71.11%, outperforming standard PT (66.43%) and SMoP (69.61%). (2) SMoP shows size-dependent behaviour: while underperforming PT on the 1B model (56.77% vs 56.25%), it outperforms PT on the 3B model (69.61% vs 66.43%). (3) PT-MoE demonstrates robust performance by outperforming the baselines on three out of five mathematical datasets. These findings collectively validate the size-independence and stability of PT-MoE across different model sizes.

5.5. Efficiency Analysis

Results in Figure 4 demonstrate PT-MoE's efficiency across both computational and parametric aspects. PT-MoE achieves the highest performance with only moderate training steps and minimal parameters (80k). In contrast, LoRA and HydraLoRA require more parameters and training steps to achieve comparable performance. Other prompt tuning methods such as PT, SMoP, and DPT converge fast but achieve lower performance, suggesting a potential trade-off between training efficiency and model effectiveness. These results validate that PT-MoE balances the computational cost, parameter efficiency, and model performance.

6. Conclusions

This work introduces PT-MoE, a novel parameter-efficient framework that integrates matrix decomposition with MoE routing for prompt tuning. Our results across 17 datasets demonstrate that PT-MoE achieves state-of-the-art performance while maintaining parameter efficiency, outperforming existing methods in both QA and mathematical tasks. Through ablation studies, we identify optimal configurations for prompt length, expert count, and routing mechanisms, providing insights for future parameter-efficient tuning approaches.

Future directions include exploring hierarchical routing mechanisms to better deal with diverse task distributions, and extending PT-MoE to continual learning scenarios for efficient adaptation and knowledge transfer across tasks.

Ethics Statement

No ethical approval was required for this study.

Availability Statement

The codes and models related to this paper are uploaded to the open-source community at https://github.com/ZongqianLi/PT-MoE.

References

Asai, A., Salehi, M., Peters, M., and Hajishirzi, H. ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6655–6672, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.446. URL https://aclanthology.org/2022.emnlp-main.446/.

Choi, J.-Y., Kim, J., Park, J.-H., Mok, W.-L., and Lee, S. SMop: Towards efficient and effective prompt tuning with sparse mixture-of-prompts. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=5x5Vxclc1K.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

Dunn, M., Sagun, L., Higgins, M., Guney, V. U., Cirik, V., and Cho, K. Searchqa: A new q&a dataset augmented with context from a search engine, 2017. URL https://arxiv.org/abs/1704.05179.

Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., and Chen, D. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*, 2019.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., ..., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lIsCS8b6zj.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Barzilay, R. and Kan, M.-Y. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Kembhavi, A., Seo, M., Schwenk, D., Choi, J., Farhadi, A., and Hajishirzi, H. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5376–5384, 2017. doi: 10.1109/CVPR.2017.571.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. MAWPS: A math word problem repository. In Knight, K., Nenkova, A., and Rambow, O. (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1157, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1136. URL https://aclanthology.org/N16-1136/.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. RACE: Large-scale ReAding comprehension dataset from examinations. In Palmer, M., Hwa, R., and Riedel, S. (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for

Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Levy, O., Seo, M., Choi, E., and Zettlemoyer, L. Zero-shot relation extraction via reading comprehension. In Levy, R. and Specia, L. (eds.), *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Li, Z. and Cole, J. M. Auto-generating question-answering datasets with domain-specific knowledge for language models in scientific tasks. *Digital Discovery*, 4(4):998–1005, 2025.

Li, Z., Su, Y., and Collier, N. 500xcompressor: Generalized prompt compression for large language models, 2024. URL https://arxiv.org/abs/2408.03094.

Li, Z., Liu, Y., Su, Y., and Collier, N. Prompt compression for large language models: A survey. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.368/.

Li, Z., Shareghi, E., and Collier, N. Reasongraph: Visualisation of reasoning paths, 2025b. URL https://arxiv.org/abs/2503.03979.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. Gpt understands, too, 2023. URL https://arxiv.org/abs/2103.10385.

Ma, F., Zhang, C., Ren, L., Wang, J., Wang, Q., Wu, W., Quan, X., and Song, D. XPrompt: Exploring the extreme of prompt tuning. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11033–11047, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.758. URL https://aclanthology.org/2022.emnlp-main.758/.

Miao, S.-y., Liang, C.-C., and Su, K.-Y. A diverse corpus for evaluating and developing English math word problem solvers. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92.

Nebrelbug. Math problems. Hugging Face Hub, 2024. URL https://huggingface.co/datasets/nebrelbug/math-problems/tree/main.

Partalas, I., Gaussier, E., Ngomo, A.-C. N., et al. Results of the first bioasq workshop. In *BioASQ@ CLEF*, pp. 1–8, 2013.

Patel, A., Bhattamishra, S., and Goyal, N. Are NLP models really able to solve simple math word problems? In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X. (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Razdaibiedina, A., Mao, Y., Khabsa, M., Lewis, M., Hou, R., Ba, J., and Almahairi, A. Residual prompt tuning: improving prompt tuning with residual reparameterization. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6740–6757, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.421. URL https://aclanthology.org/2023.findings-acl.421/.

Saha, A., Aralikatte, R., Khapra, M. M., and Sankaranarayanan, K. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1683–1693, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1156. URL https://aclanthology.org/P18-1156.

Shi, Z. and Lipani, A. DePT: Decomposed prompt tuning for parameter-efficient fine-tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KjegfPGRde.

Vu, T., Lester, B., Constant, N., Al-Rfou', R., and Cer, D. SPoT: Better frozen model adaptation through soft prompt transfer. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5039–5059, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.346. URL https://aclanthology.org/2022.acl-long.346/.

Wang, C., Wang, J., Qiu, M., Huang, J., and Gao, M. TransPrompt: Towards an automatic transferable prompting framework for few-shot text classification. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2792–2802, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.221. URL https://aclanthology.org/2021.emnlp-main.221/.

Wang, Z., Panda, R., Karlinsky, L., Feris, R., Sun, H., and Kim, Y. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Nk2pDtuhTq.

Xiao, Y., Xu, L., Li, J., Lu, W., and Li, X. Decomposed prompt tuning via low-rank reparameterization. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13335–13347, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.890. URL https://aclanthology.org/2023.findings-emnlp.890/.

Xu, Y., Zeng, Z., and Shen, Z. Efficient cross-task prompt tuning for few-shot conversational emotion recognition. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11654–11666, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.780. URL https://aclanthology.org/2023.findings-emnlp.780/.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.