LoPT: Low-Rank Prompt Tuning for Parameter Efficient Language Models

Anonymous ACL submission

Abstract

In prompt tuning, a prefix or suffix text is added to the prompt, and the embeddings (soft prompts) or token indices (hard prompts) of the prefix/suffix are optimized to gain more control over language models for specific tasks. This approach eliminates the need for hand-crafted prompt engineering or explicit model fine-tuning. Prompt tuning is significantly more parameter-efficient than model fine-tuning, as it involves optimizing partial inputs of language models to produce desired outputs.

In this work, we aim to further reduce the amount of trainable parameters required for a language model to perform well on specific tasks. We propose Low-rank Prompt Tuning (LoPT), a low-rank model for prompts that achieves efficient prompt optimization. The proposed method demonstrates similar outcomes to full parameter prompt tuning while reducing the number of trainable parameters by a factor of 5. It also provides promising results compared to the state-of-the-art methods that would require 10 to 20 times more parameters.

1 Introduction

With the success of large language models (Touvron et al., 2023; Achiam et al., 2023; Jiang et al., 2023), it has become increasingly important for language models (LMs) to handle instructions effectively for customized agents and tasks. There are three essential categories of methods to adapt pretrained language models to specific and customized needs: prompt engineering, model fine-tuning, and prompt tuning.

Prompt engineering (Brown et al., 2020; Sanh et al., 2021; Chung et al., 2024) involves crafting handcrafted prompts and faces the challenge of getting LMs to consistently produce desired outputs with few-shot instructions. This effort may be difficult to generalize or extend to new tasks. Model fine-tuning (Raffel et al., 2020) can perform very well for task-specific needs but requires explicit fine-tuning of a significant number of model parameters, even with parameter-efficient fine-tuning (PEFT) approaches (Liu et al., 2022; Hu et al., 2021).

Prompt tuning (PT) (Li and Liang, 2021; Lester et al., 2021; Wen et al., 2024; Shi et al., 2022; Shin et al., 2020; Khashabi et al., 2021) is a promising method that lies between prompt engineering and model fine-tuning. Instead of handcrafting prompts, it optimizes a small number of prompt embeddings or indices with training data and has demonstrated capabilities comparable to those of model fine-tuning approaches (Asai et al., 2022; Shi and Lipani, 2023; Wang et al., 2023).

We focus on soft prompt tuning, which operates by adding a prefix or suffix to the existing inputs and optimizing the embeddings of this prefix or suffix. The embeddings, or the soft prompt matrix, has dimensions $n \times d$, where n is the “tokens" length of soft prompts, and d is the embedding size. The soft prompt length n can be task specific to achieve desired outcomes. For example, more sophisticated tasks might benefit from longer soft prompts that allow for more parameters to be optimized.

In this work, we introduce a low-rank modeling approach for the soft prompt matrix, which effectively reduces the number of trainable parameters in prompt tuning without compromising performance. We find that soft prompt matrices are inherently low-rank due to their dimensionality, and we apply further dimensionality reduction through our proposed method. We demonstrate that the number of parameters required for tuning LMs to meet specific task requirements can be minimal. Additionally, the number of trainable parameters can be easily controlled by adjusting the rank of the soft prompt matrix.

Our approach distinguishes itself from existing methods by directly imposing low-rank constraints on the entire soft prompt to be trained. While recent work (Shi and Lipani, 2023) also explores low-rank matrices for prompt tuning, it restricts low-rankness to the differences or updates of a frozen baseline prompt, similar to the LoRA technique used in model fine-tuning (Hu et al., 2021), and is only applied to a portion of the overall soft prompt.

Our primary contributions are:

- We introduce Low-rank Prompt Tuning (LoPT) that significantly reduces the number of trainable parameters required in prompt tuning.
- We achieve a 5-fold reduction in trainable parameters while maintaining performance comparable to the full-parameter prompt tuning.
- We demonstrate the efficacy of our method across 5 diverse datasets, showing substantial improvements in parameter efficiency compared to existing methods.

Our proposed parameter-efficient method would be particularly beneficial for computationally demanding prompt tuning needs in sophisticated tasks and large language models.

2 Method

2.1 Problem statement

In soft prompt tuning (Lester et al., 2021), we add a prefix or suffix to the original prompt and optimize the embeddings of this prefix or suffix as trainable parameters using supervised training data to achieve task-specific predictions.

Given a language model $\mathcal{M}$ with frozen network parameters $\boldsymbol{\theta}$ and embedding matrix $\boldsymbol{E} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size, d is the embedding size, with each row of $\boldsymbol{E}$ representing a token in the vocabulary. We optimize trainable embeddings $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ of the prefix, where n is the number of soft tokens. The optimization problem can be formulated as:

$$\arg\min_{\boldsymbol{X}} \sum_i \mathcal{L}\left(\mathcal{M}\left([\boldsymbol{X};\boldsymbol{I}_i];\boldsymbol{\theta}\right), \boldsymbol{y}_i\right), \quad (1)$$

where $\mathcal{L}$ is the loss function for the task. For the i-th training sample, $\boldsymbol{I}_i \in \mathbb{R}^{t \times d}$ denotes tokenized embeddings of the original model input with sequence length t, and $\boldsymbol{y}_i$ is the label associated with this sample.

2.2 Our Low-Rank Prompt Tuning (LoPT)

Recent work (Lester et al., 2021; Shi and Lipani, 2023) demonstrates that prompt tuning could yield performance comparable to parameter-efficient model fine-tuning methods (Hu et al., 2021) with a significantly smaller amount of learnable parameters. In this work, we push the boundaries by exploring parameter-efficient prompt tuning to further reduce the number of trainable parameters without compromising accuracy.

Because the prefix or suffix length n is often significantly smaller that the embedding dimension d in prompt tuning, the rank of the soft prompt matrix $\boldsymbol{X}$ would inherently be constrained by n, making $\boldsymbol{X}$ low-rank. The potential similarity between neighboring embeddings in a prompt could also suggest that $\boldsymbol{X}$ is low-rank. Therefore, we explore this potential and impose constraints on $\boldsymbol{X}$ for dimensionality reduction and more efficient prompt tuning.

We propose two low-rank approximations for modeling $\boldsymbol{X}$. The proposed methods could drastically reduce the number of learnable parameters while maintaining performance comparable to full-parameter prompt tuning.

2.2.1 LoPT-1

For effective prompt tuning with a reduced and adjustable number of parameters, we propose to decomposite the low-rank prompt matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ as:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}. \quad (2)$$

In this formulation, $\boldsymbol{U} \in \mathbb{R}^{n \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times d}$ are the new trainable matrices. We train $\boldsymbol{U}$ and $\boldsymbol{V}$ simultaneously, transforming the prompt tuning optimization problem to the following:

$$\arg\min_{\boldsymbol{U},\boldsymbol{V}} \sum_i \mathcal{L}\left(\mathcal{M}\left([\boldsymbol{U}\boldsymbol{V};\boldsymbol{I}_i];\boldsymbol{\theta}\right), \boldsymbol{y}_i\right). \quad (3)$$

We initialize both $\boldsymbol{U}$ and $\boldsymbol{V}$ with uniform random values in the range of [-0.5, 0.5] at the beginning of training.

The number of trainable parameters is reduced to $r(n+d)$. As $n \ll d$, the total number of parameters can be significantly reduced compared to the original nd, especially with adjustable choices of $r < n$.

2.2.2 LoPT-2

we also introduce an empirical mapping scheme for the low-rank approximation of $\boldsymbol{X}$, employing

Method	# Params	SST-2	AGNews
No LoPT	12.8k	92.8	91.8
LoPT-1 (ours)	2.58k	92.1	91.9
LoPT-2 (ours)	5.12k	90.9	90.0

Table 1: Accuracy (%) on the SST-2 and AGNews validation sets compares the proposed LoPT-1 and LoPT-2 to the baseline soft prompt tuning without low-rank factorization (No LoPT). The language model used is GPT-2 large with embedding dimension $d = 1280$, and prompt length $n = 10$. We set the rank $r = 2$ for both LoPT-1 and LoPT-2, and calculate the # of parameters accordingly.

learnable linear projections and nonlinear thresholding operation to achieve effects analogous to singular value thresholding (Cai et al., 2010) and with reduced number of parameters for optimization. Specifically, we construct $\boldsymbol{X}$ as:

$$\boldsymbol{X} = \sigma(\boldsymbol{X}_0 \boldsymbol{U})\boldsymbol{V}, \tag{4}$$

where $\boldsymbol{X}_0 \in \mathbb{R}^{n\times d}$ is a random initialization of $\boldsymbol{X}$, $\boldsymbol{U} \in \mathbb{R}^{d\times r}$ and $V \in \mathbb{R}^{r\times d}$ are linear projection matrices. $\sigma(\cdot) = \max(\cdot, 0)$ represents the nonlinear thresholding operation that filters out negative values. Similar to LoPT-1, $\boldsymbol{U}$ and $\boldsymbol{V}$ are randomly initialized and optimized with function

$$\underset{\boldsymbol{U},\boldsymbol{V}}{\arg\min} \sum_i \mathcal{L}\left(\mathcal{M}\left([\sigma(\boldsymbol{X}_0\boldsymbol{U})\boldsymbol{V}; \boldsymbol{I}_i]; \boldsymbol{\theta}\right), \boldsymbol{y}_i\right). \tag{5}$$

The number of trainable parameters becomes $2rd$ rather than nd. By choosing a smaller projected dimension $r < n/2$, we can easily reduce redundancy in trainable parameters and improve time and memory efficiency. It is worth noting that for $n \ll d$, LoPT-1 is more parameter efficient than LoPT-2.

Implementation Simplification The proposed LoPT-2 mapping for $\boldsymbol{X}$ improves parameter efficiency, and we propose a straightforward implementation. We use two linear layers for the linear projections U and V, and apply an ELU (Clevert et al., 2015) function for the nonlinear thresholding operator $\sigma(\cdot)$. Empirically, we found that ELU performs better than ReLU (Nair and Hinton, 2010; Fukushima, 1969) and GELU (Hendrycks and Gimpel, 2016).

We demonstrate that the proposed low-rank modeling and formulations yield effective parameter reduction with promising outcomes.

3 Experiments

3.1 Experiment Setup

Datasets We evaluate the proposed method on classification tasks using various datasets in English: the sentiment analysis task SST-2 (Socher et al., 2013), the 4-way topic classification task AGNews (Zhang et al., 2015), and datasets in the SuperGLUE benchmark (Wang et al., 2019). These include BoolQ (Clark et al., 2019), RTE (Giampiccolo et al., 2007), WiC (Pilehvar and Camacho-Collados, 2018), and CB (De Marneffe et al., 2019).

Training Details The proposed low-rank factorizations, LoPT-1 and LoPT-2, are optimized using GPT-2 large (774M parameters, $d = 1280$) (Radford et al., 2019) and T5-base (220M parameters, $d = 768$) (Raffel et al., 2020) models. We build upon the settings in (Ding et al., 2021; Wen et al., 2024), and optimize the prompts using the Adafactor optimizer (Shazeer and Stern, 2018) with a learning rate of 0.3. We apply soft prompt length n of 10 or 20, and batch size of 8 for SuperGLUE datasets, and 16 for other data.

We set the rank parameter r of LoPT-1 or LoPT-2 to $\lfloor \frac{n}{4} \rfloor$ for most experiments to achieve the desired level of trainable parameter reduction. In the case of prompt tuning without our proposed low-rank approximations, the number of trainable parameters is nd. For LoPT-1, the number of learnable parameters is $r(n + d)$. For LoPT-2, the trainable parameter amount is $2dr$.

3.2 Comparisons and Results

We compare the proposed parameter efficient approaches to vanilla soft prompt tuning using the GPT-2 large model, and evaluate their effectiveness with SST-2 and AGNews datasets. As presented in Table 1, LoPT-1 significantly reduces the number of trainable parameters from 12.8k to 2.58k, while maintaining accuracy levels comparable to full parameter prompt tuning. LoPT-2 achieves a 60% reduction in parameters and successfully preserves classification accuracy for both binary and multi-class classification tasks.

Our methods are compared against a variety of baselines including Fine-tuning, LoRA (Hu et al., 2021), PT (Lester et al., 2021), and DePT (Shi and Lipani, 2023) using the T5-base model. As shown in Table 2, LoPT-1 and LoPT-2 demonstrate promising performance, achieving reductions in trainable parameters by factors of 20 and 10, respectively.

Method	# Params	SST-2	BoolQ	RTE	WiC	CB
Fine-tuning[1]	220M	94.6	81.1	71.9	70.2	85.7
LoRA[2]	3.8M	94.3	81.3	75.5	68.3	92.9
PT[3]	76.8k	91.9	63.7	78.8	50.8	67.9
DePT[3]	76.8k	94.2	79.3	79.1	68.7	92.9
LoPT-1 (ours)	3.94k	92.9	76.5	73.8	55.1	90.4
LoPT-2 (ours)	7.68k	92.4	75.5	74.3	62.7	74.0

Table 2: Accuracy (%) on the SST-2 and SuperGLUE benchmarks for classification tasks. The language model is T5-Base with embedding dimension $d = 768$. We set the rank $r = 5$ and soft prompt length $n = 20$ for both LoPT-1 and LoPT-2. Comparisons including Fine-tuning[1] from (Asai et al., 2022), LoRA[2] from (Sung et al., 2022), PT[3] and DePT[3] are from (Shi and Lipani, 2023).

Length	Rank	Δ # Params	SST-2
$n = 10$	No LoPT	-	92.8
	$r = 1$	-89.92%	90.5
$n = 10$	$r = 2$	-79.84%	92.1
	$r = 5$	-49.61%	92.1
	$r = 1$	-89.84%	91.4
$n = 20$	$r = 2$	-79.69%	92.8
	$r = 5$	-49.22%	**92.9**
	$r = 1$	-89.77%	90.9
$n = 30$	$r = 2$	-79.53%	92.2
	$r = 5$	-48.83%	92.1

Table 3: Ablation study on LoPT-1: We evaluated various combinations of prompt length n and rank r using the SST-2 dataset and the GPT-2 large model. The numbers of trainable parameters are compared to the baseline prompt tuning, which has a fixed $n = 10$ and no low-rank approximations. The parameter reduction rate is represented by Δ # Params. LoPT-1 with $n = 20$ and $r = 5$ achieves the highest accuracy (%).

This marks a significant efficiency improvement over existing prompt tuning approaches, which are already noted for their high parameter efficiency.

It is noteworthy that LoPT-1 outperforms LoPT-2 on the CB dataset, while LoPT-2 excels over LoPT-1 on the WiC dataset. This suggests that both approaches could be strategically exploited to tailor the desired low-rank formation for optimal performance on specific tasks.

3.3 Ablation Study

Using the SST-2 task and the GPT-2 large model, Table 3 presents the accuracy of LoPT-1 with varying prompt lengths n and ranks r for the low-rank factorization. We observe that an increased prompt length does not necessarily lead to improved outcomes, and the combination of $n = 20$ with $r = 5$ or $r = 2$ yield the highest accuracy. Given that n is much smaller than d, the number of trainable parameters is primarily controlled by the rank parameter r in LoPT, which can be easily adjusted to achieve parameter reduction.

3.4 Limitations

This work relies on the low-rank hypothesis and may not be effective when the prompt matrix is not low-rank. Regarding the performance of the proposed methods, further improvements could be achieved through hyper-parameter tuning.

4 Conclusion

In this work, we propose Low-rank Prompt Tuning (LoPT), a low-rank formulation of prompts that significantly reduces the number of trainable parameters for parameter-efficient prompt tuning of language models. We demonstrate that LoPT can decrease the number of trainable parameters by a factor of 10 or 20 while achieving promising performance across various datasets.

The proposed parameter-efficient method could be particularly beneficial for sophisticated tasks and large language models, where longer soft prompts are increasingly important for effective prompt tuning.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. *arXiv preprint arXiv:2205.11961*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Kunihiko Fukushima. 1969. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

Daniel Khashabi, Shane Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, et al. 2021. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. *arXiv preprint arXiv:2112.08348*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Weijia Shi, Xiaochuang Han, Hila Gonen, Ari Holtzman, Yulia Tsvetkov, and Luke Zettlemoyer. 2022. Toward human readable prompt tuning: Kubrick's the shining is a good movie, and a good prompt too? *arXiv preprint arXiv:2212.10539*.

Zhengxiang Shi and Aldo Lipani. 2023. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *arXiv preprint arXiv:2309.05173*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. pages 1631–1642, Seattle, Washington, USA.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35:12991–13005.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023. Multitask prompt tuning enables parameter-efficient transfer learning. *arXiv preprint arXiv:2303.02861*.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.